MATEMATICA PER INFORMATICA

A. INTRODUZIONE

In questa sezione raccogliamo le parti della matematica che sono utilizzate ed applicate in informatica.
Intendiamoci: non ci occuperemo di parti esclusive di informatica come, ad esempio la gestione dei numeri da parte del computer, ma solamente di quelle parti effettivamente appartenenti alla matematica, come ad esempio la differenza calcolata come somma al complementare, che vengono poi utilizzate in informatica questi gli argomenti che tratteremo:

- Sistemi di numerazione
- Cenni sull'algebra binaria di Boole
- Circuiti logici ed elettrici
- Algoritmi?
- Automi?

B. SISTEMI DI NUMERAZIONE

1. Contare

Fin dall'antichita' l'uomo si deve essere posto il problema di "contare" cioe' di sapere quanto di qualcosa fosse disponibile, magari per motivi di sopravvivenza; sembra che gia', in un osso di lupo, datato 30.000 anni fa, qualcuno abbia segnato delle tacche in modo regolare raggruppandole in gruppi di 5.
Il problema era come contare se gli oggetti erano molti: finche' si era a numeri inferiori al numero delle dita delle mani non c'erano problemi, ma con numeri piu' grandi? Da qui la grande idea di raggruppare e ricominciare da capo, ad esempio:
1 dito
2 dita
3 dita
4 dita
5 dita = 1 mano
1 mano + 1 dito
1 mano + 2 dita
....
E che succede se il numero e' veramente grande? Semplicemente bastera' fare un gruppo fra i gruppi potendo cosi' continuare all'infinito:
4 mani + 4 dita
4 mani + 5 dita = 5 mani = 1 manodimani (nuovo gruppo)
1 manodimani + 1 dito
1 manodimani + 2 dita
1 manodimani + 3 dita
1 manodimani + 5 dita
1 manodimani + 1 mano
1 manodimani + 1 mano + 1 dito
1 manodimani + 1 mano + 2 dita
....

Questo ragionamento e' alla base dei **sistemi di numerazione** che sono quelli che decidono come fare i gruppi, i gruppi dei gruppi, i gruppi dei gruppi dei gruppi, e quindi ci permettono di poter contare finche' vogliamo.

2. Cos'e' un sistema di numerazione

Un sistema di numerazione e' ogni complesso di regole e simboli che servano per scrivere e leggere i numeri.
Siccome noi, normalmente, contiamo a base 10, cioe' raggruppiamo ogni 10, avremo, nel sistema di numerazione decimale:
10 simboli 0,1,2,3,4,5,6,7,8,9 chiamati cifre,
una regola "posizionale", cioe' la regola che:
Ogni cifra ha valore secondo la sua posizione.
Esempio:
131 nel nostro sistema di numerazione decimale significa che, partendo da sinistra, il primo 1 rappresenta un gruppo di 10 al secondo ordine (centinaia) mentre il terzo 1 rappresenta una cifra e basta (gruppo di ordine 0 od unita').
E' possibile utilizzare un qualunque insieme di cifre superiore ad 1 per costruire un qualunque sistema di numerazione; cio' e' dovuto al fatto che possiamo sempre rappresentare un numero in **forma polinomiale**.

Forma polimoniale: ogni numero e' rappresentabile come
$\mathbf{a \cdot x^n + b \cdot x^{n-1} + \ldots + c \cdot x^2 + d \cdot x + e}$
o meglio:
$\mathbf{a \cdot x^n + b \cdot x^{n-1} + \ldots + c \cdot x^2 + d \cdot x^1 + e \cdot x^0}$
con **x** base del sistema di numerazione e **a, b, c, d, e, f, ..** cifre che possono anche ripetersi (al massimo possono essere **x**).
Ricordo che $\mathbf{x^1 = x}$ e $\mathbf{x^0 = 1}$

Cosi', se voglio rappresentare il numero dato dalle seguenti tacche (le metto a gruppi di 5 per farvele vedere meglio, ma dovrebbero essere senza spazi)
///// ///// ///// ///// ///// ///// ///// ///// ///// ///// ///// ///// ///// ///

- **In forma decimale**

Posso usare le cifre **0, 1, 2, 3, 4, 5, 6, 7, 8, 9**
Raggruppo a gruppi di 10 con le parentesi tonde, ottengo:
(///// /////) (///// /////) (///// /////) (///// /////) (///// /////) (///// /////)
///// ///
Ho 6 raggruppamenti da 10 e 8 non raggruppati cioe' $\mathbf{68_{10}}$ od anche, in forma polinomiale
$\mathbf{68_{10} = 6 \cdot 10 + 8}$ cioe' 6 decine ed 8 unita'.

- **In base 5**

Posso usare solo le cifre **0, 1, 2, 3, 4**
raggruppo a gruppi di 5 con le parentesi tonde, ottengo
(/////) (/////) (/////) (/////) (/////) (/////) (/////) (/////) (/////) (/////) (/////)
(/////) (/////) ///
Siccome i termini raggruppati sono superiori a 4 devo fare un loro raggruppamento 5 a 5, (chiamiamole venticinquine) lo indico con la parentesi quadra:
[(/////) (/////) (/////) (/////) (/////)] [(/////) (/////) (/////) (/////) (/////)]
(/////) (/////) (/////)
///
Ho 2 raggruppamenti del secondo ordine, 3 raggruppamenti del primo ordine e 3 non

raggruppati, cioe' **233_5** od anche, in forma polinomiale:
$233_5 = 2 \cdot 5^2 + 3 \cdot 5 + 3$ cioe' 2 venticinquine, piu' 3 cinquine, piu' 3 unita'

- **In base 3**

Posso usare solo le cifre **0, 1, 2**
Raggruppo a gruppi di 3, ottengo:
(///) (///) (///) (///) (///) (///)/(///) (///) (///) (///) (///) (///) (///) (///) (///) (///) (///) (///) (///) (///) (///) (///)/
Siccome i termini raggruppati sono superiori a 2, devo fare un loro raggruppamento 3 a 3, (diciamoli gruppi 3^2) lo indico con la parentesi quadra:
[(///) (///) (///)] [(///) (///) (///)] [(///) (///) (///)] [(///) (///) (///)] [(///) (///) (///)] [(///) (///) (///)] [(///) (///) (///)] (///) //
Siccome i termini con parentesi quadra raggruppati sono superiori a 2 devo fare ancora un loro raggruppamento 3 a 3, (diciamoli gruppi 3^3) lo indico con la parentesi graffa:
{[(///) (///) (///)] [(///) (///) (///)] [(///) (///) (///)]} {[(///) (///) (///)] [(///) (///) (///)] [(///) (///) (///)]}
[(///) (///) (///)]
(///)
//
Ho 2 raggruppamenti del terzo ordine (3^3), 1 raggruppamento del secondo ordine (3^2), 1 raggruppamento del primo ordine e 2 non raggruppati, cioe' **2112_3** od anche, in forma polinomiale:
$2112_3 = 2 \cdot 3^3 + 1 \cdot 3^2 + 1 \cdot 3 + 2$ cioe' 2 gruppi 3^3 piu' un gruppo 3^2 piu' una terna piu' 2 unita'.
Notiamo che, piu' abbassiamo la base, piu' diventa difficile per noi leggere il numero e complicato lo scriverlo: se pero' riusciamo a utilizzare solo due cifre (0 ed 1) allora il numero sara' ancora piu' complicato, ma potremo farlo leggere ad una macchina, associando magari lo 0 alla mancanza di corrente e l'1 al passaggio di corrente e poi saranno problemi della macchina dover trattare numeri complicati (tanto le macchine non possono protestare)

3. Numeri in forma polinomiale

In questa pagina vediamo come, matematicamente, partendo da un numero come siamo abituati a trattarlo (forma decimale), e' possibile trasformarlo in forma polinomiale:
$a \cdot x^n + b \cdot x^{n-1} + \ldots + c \cdot x^2 + d \cdot x^1 + e \cdot x^0$
e quindi poterlo scrivere in un altra unita' di misura.
In pratica formalizziamo matematicamente quanto fatto nella pagina precedente sul numero:
// = **68_{10}**
utilizzando la forma 68_{10} per semplicita'.

Per trasformare un numero decimale in altra base si deve dividere tale numero per la base scelta riportando i resti e ripetendo il procedimento sinche' il resto e' minore della base scelta: la lettura dei resti partendo dall'ultimo sino al primo mi fornisce il numero nella base cercata.

E' lo stesso procedimento della pagina precedente. Se prendo il numero:
//
e raggruppo i termini per 10 e' come se lo divido per 10 ed ottengo come resto gli ultimi termini (quelli che non entrano in una parentesi):

(//////////) (//////////) (//////////) (//////////) (//////////) (//////////) ////////
Ho quindi come resto **8.**
Divido di nuovo, stavolta le parentesi per 10 e siccome sono 6 cioe' meno di 10 ottengo come resto **6;** quindi metto in ordine inverso i due resti ed ottengo:
68_{10}

- **In base 5**
  Nel risultato posso usare solo le cifre **0, 1, 2, 3, 4**
  Come ho gia' detto prendiamo 68_{10} per comodita', ma potremmo farlo tranquillamente sul numero:
  //
  dividendo per 5, cioe' raggruppando ricorsivamente per gruppi di 5 (raggruppare e' equivalente a dividere)

  Divido il numero 68 per 5
  $68:5 \Rightarrow \; 68 = 5 \cdot 13 + 3$ cioe' ottengo quoziente 13 e **resto = 3 (primo resto)**
  Ripeto il procedimento sul 13
  $13:5 \Rightarrow \; 13 = 5 \cdot 2 + 3$ cioe' ottengo quoziente 2 e **resto = 3 (secondo resto)**
  Ripeto il procedimento sul 2
  $2:5 \Rightarrow \; 2 = 5 \cdot 0 + 2$ cioe' ottengo quoziente 0 e **resto = 2 (ultimo resto)**

  Ordino i resti dall'ultimo al primo ed ottengo 233_5 od anche, in forma polinomiale:
  $233_5 = 2 \cdot 5^2 + 3 \cdot 5^1 + 3 \cdot 5^0 = 2 \cdot 5^2 + 3 \cdot 5 + 3$
  Se lo rivoglio in forma decimale, basta sviluppare le potenze e calcolare:
  $2 \cdot 5^2 + 3 \cdot 5 + 3 = 2 \cdot 25 + 3 \cdot 5 + 3 = 50 + 15 + 3 = 68$

- **In base 3**
  Nel risultato posso usare solo le cifre **0, 1, 2**
  Divido il numero 68 per 3:
  $68:3 \Rightarrow \; 68 = 22 \cdot 3 + 2$ cioe' ottengo quoziente 22 e **resto = 2 (primo resto)**
  Ripeto il procedimento sul 22:
  $22:3 \Rightarrow \; 22 = 3 \cdot 7 + 1$ cioe' ottengo quoziente 7 e **resto = 1 (secondo resto)**
  Ripeto il procedimento sul 7:
  $7:3 \Rightarrow \; 7 = 3 \cdot 2 + 1$ cioe' ottengo quoziente 2 e **resto = 1 (terzo resto)**
  Ripeto il procedimento sul 2:
  $2:3 \Rightarrow \; 2 = 3 \cdot 0 + 2$ cioe' ottengo quoziente 0 e **resto = 2 (ultimo resto)**

  Ordino i resti dall'ultimo al primo ed ottengo 2112_3 od anche, in forma polinomiale:
  $2112_3 = 2 \cdot 3^3 + 1 \cdot 3^2 + 1 \cdot 3^1 + 2 \cdot 3^0 = 2 \cdot 3^3 + 1 \cdot 3^2 + 1 \cdot 3 + 2$
  Se lo rivoglio in forma decimale basta sviluppare le potenze e calcolare
  $2 \cdot 3^3 + 1 \cdot 3^2 + 1 \cdot 3 + 2 = 2 \cdot 27 + 1 \cdot 9 + 1 \cdot 3 + 2 = 54 + 9 + 3 + 2 = 68$

4. Numeri e sistemi di numerazione nell'antichita'

Facciamo un po' di storia; partiamo dall'osso di lupo di 30.000 anni fa, di cui abbiamo parlato, in cui vi sono delle tacche raggruppate a gruppi di 5 che fanno pensare ad un primo sistema di calcolo a base 5.

Nella lingua francese abbiamo i numeri:
....
settantanove *septante neuf*

ottanta *quatre vingt* (quattro venti)
ottantuno *quatre vingt et un* (quattro venti e uno)
....
ottantanove *quatre vingt et neuf* (quattro venti e nove)
novanta *quatre vingt dix* (quattro venti dieci)
novantuno *quatre vingt et onze* (quattro venti e undici)
....
novantanove *quatre vingt dixneuf* (quattro venti diciannove)
cento *cent*
....
che fanno pensare ad un sistema arcaico basato sul 20.

I primi di cui abbiamo notizie certe sono gli egiziani che gia' 8000 anni fa usavano un sistema decimale, ma avevano grossi problemi con le frazioni.
I babilonesi risolsero il problema delle frazioni considerando un sistema a base 60 (la base 60 e' divisibile per 2,3,4,5,6,10,12,15,20,30,60 quindi l'utilizzo delle frazioni e' molto facilitato). Ancora oggi, come eredita' dei babilonesi, misuriamo il tempo suddividendo le ore in 60 minuti ed i minuti in 60 secondi:
....
cinquantotto minuti
cinquantanove minuti
sessanta minuti = 1 ora
1 ora ed un minuto
....
Inoltre misuriamo un angolo giro mediante 360 gradi ed ogni grado si puo' suddividere in 60 primi ed ogni primo in 60 secondi.
Come gli egiziani anche i romani utilizzavano un sistema a base 10, ed anch'essi non utilizzavano un sistema "posizionale", cioe' un numero aveva sempre lo stesso valore in qualunque ordine si scrivessero le sue cifre

---

Da questo fatto deriva che si pensa che il numero 17 porti sfortuna, infatti i romani, scrivendolo in modo disordinato, come ho fatto qui a destra
potevano leggere "Vixi" cioe' "ho vissuto" che equivale a dire "sono morto"

V I I X

Occorre arrivare al medio evo per avere il sistema decimale posizionale, cioe' una cifra ha valore (di decina, di centinaia, di migliaia,....) a seconda del posto che occupa; nell'esempio successivo il 2 ha un valore diverso nei due numeri considerati: nel primo vale 200 nel secondo vale solamente due unita':

$\mathbf{1200 = 1 \cdot 10^3 + 2 \cdot 10^2 + 0 \cdot 10^1 + 0 \cdot 10^0 = 1000+200+0+0}$

$\mathbf{1002 = 1 \cdot 10^3 + 0 \cdot 10^2 + 0 \cdot 10^1 + 2 \cdot 10^0 = 1000+0+0+2}$

5. Il sistema di numerazione decimale

Vediamo in questa pagina di mettere assieme quanto detto sull sistema di numerazione decimale.
Il sistema decimale si basa sul **10;** cioe', partendo dalle unita', quando arrivo al decimo oggetto, raggruppo e passo all'ordine superiore: decine, centinaia, migliaia. Essendo basato sul 10, ho bisogno di 10 cifre per poter indicare le diverse unita' :
0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Naturalmente tali cifre serviranno ad indicare anche il numero dei gruppi di 10 (o decine), il numero dei gruppi dei gruppi di 10 (10 · 10 o centinaia). Ad esempio:
25 = 2 gruppi di 10 + 3 unita' = 2 decine + 3 unita' = $2\cdot 10 + 3$
325 = 3 gruppi di cento + 2 gruppi di dieci + 3 unita' = $3\cdot 10\cdot 10 + 2\cdot 10 + 5 = 3\cdot 10^2 + 2\cdot 10^1 + 5\cdot 10^0$
Ricordando che: **$10^1 = 10$** e **$10^0 = 1$**
6325481 = 6 milioni + 3 centinaia di migliaia + 2 decine di migliaia + 5 migliaia + 4 centinaia + 8 decine + 1 unita' = $6\cdot 10^6 + 3\cdot 10^5 + 2\cdot 10^4 + 5\cdot 10^3 + 4\cdot 10^2 + 8\cdot 10^1 + 1\cdot 10^0$

Il sistema e' ***"posizionale"***, cioe' una cifra, ha valore diverso a seconda del posto che occupa nel numero e precisamente, considerando il numero nel suo insieme:
partendo da destra abbiamo la cifra delle unita'
la seconda cifra (sempre partendo da destra) sara' la cifra delle decine
la terza cifra (sempre partendo da destra) sara' la cifra delle centinaia
la quarta cifra (sempre partendo da destra) sara' la cifra delle migliaia
la quinta cifra (sempre partendo da destra) sara' la cifra delle decine di migliaia
la sesta cifra (sempre partendo da destra) sara' la cifra delle centinaia di migliaia
....
Quindi ad esempio nel numero **2132**
il primo numero a destra significa 2 unita' **2**
il quarto numero partendo da destra significa 2 migliaia **2000**

Quando il numero decimale viene espresso come somma di potenze del 10 e' detto in forma polinomiale :
$5481 = 5\cdot 10^3 + 4\cdot 10^2 + 8\cdot 10^1 + 1\cdot 10^0$
e' in forma polinomiale

Nel numero possiamo anche considerare i decimi delle unita', i centesimi delle unita' i millesimi delle unita'..... ricordando che per la forma esponenziale vale:

$$\frac{1}{10} = \frac{1}{10^1} = 10^{-1} \qquad \frac{1}{100} = \frac{1}{10^2} = 10^{-2} \qquad \frac{1}{1000} = \frac{1}{10^3} = 10^{-3}$$

Ad esempio il numero **324,567** in forma esponenziale sara'
$324{,}567 = 3\cdot 10^2 + 2\cdot 10^1 + 4\cdot 10^0 + 5\cdot 10^{-1} + 6\cdot 10^{-2} + 7\cdot 10^{-3}$

Invece di dire migliaia di migliaia di migliaia, oppure un milione di miliardi od un millesimo di milionesimo..., definiamo dei prefissi (multipli e sottomultipli) per numeri molto grandi e molto piccoli.

Multiplo	Simbolo	Valore		Sottomultiplo	Simbolo	Valore
deca decina	da	10^1		decimo	d	10^{-1}
centinaio etto	h	10^2		centesimo	c	10^{-2}
kilo mille	k	10^3		millesimo	m	10^{-3}
Mega milione	M	10^6		micro milionesimo	µ	10^{-6}
Giga miliardo	G	10^9		nano miliardesimo	n	10^{-9}
Tera	T	10^{12}		pico	p	10^{-12}
Peta	P	10^{15}		femto	f	10^{-15}
. . .	. . .	. . . . .		. . . .	. . . .	. . . . .

6. Sistema binario

a) Introduzione

Il sistema binario, fu pensato, come idea matematica senza applicazioni pratiche, gia' nel diciassettesimo secolo, ma fu soltanto con Boole e poi con la nascita dell'informatica che acquisto' l'importanza che oggi gli compete, quale sistema di numerazione ormai piu' utilizzato nel mondo.
Intuitivamente possiamo dire che la sua importanza deriva dal fatto che e' possibile far "ricordare" ad un circuito che e' stato percorso da una corrente, ad esempio facendo attivare, al passaggio di corrente, una piccola "calamita" e quindi possiamo associare la cifra **0** al fatto della calamita non attivata e la cifra **1** alla calamita attivata. Cosi' e' possibile avere una memoria che un'altra corrente, passando per il circuito potra' leggere e magari, con un circuito opportuno che operi sui dati che incontra, trasformare in altri dati.

b) La "tabellina" della somma nei numeri binari

Il sistema binario si basa su raggruppamenti di 2 oggetti, quindi potremo utilizzare solamente due cifre, **0** ed **1.** La tabella per eseguire le operazioni di somma (come la tavola pitagorica per la somma) e' la seguente:

+	0	1
0	0	1
1	1	10

Cioe'

$0 + 0 = 0$
$0 + 1 = 1$
$1 + 0 = 1$
$1 + 1 = 10$

Quando, nel sistema decimale eseguo la somma 9+1 scrivo 0 e riporto un 1 nella colonna delle decine 9+1=10, lo stesso, nel sistema binario, quando eseguo la somma 1+1 scrivo 0 e riporto 1 nella colonna delle coppie 1+1=10: nel sistema binario, per le operazioni, la coppia corrisponde alla decina nel sistema decimale.
A sinistra, la somma 1+1=10 col riporto in verde per il sistema binario ed a destra la somma 9+1=10 sempre con il riporto in verde per il sistema decimale:

1←	
1	+
1	=
10	

1←	
9	+
1	=
10	

c) I numeri binari

Costruiamo una tabella trovando i numeri binari corrispondenti ai numeri decimali:

Decimale	numero successivo	binario
0	0	0
1	0 + 1 = 1	1
2	1 + 1 = 10	10
3	10 + 1 = 11	11
4	11 + 1 = 100	100
5	100 + 1 = 101	101
6	101 + 1 = 110	110
7	110 + 1 = 111	111
8	111 + 1 = 1000	1000
9	1000 + 1 = 1001	1001
10	1001 + 1 = 1010	1010
......	....	....

Costruisco la tabella semplicemente contando (1+1+1...).
Costruiamola assieme fino al numero 10, poi e' sempre la stessa cosa; aggiungiamo **1** ogni volta e calcoliamo il risultato:

0. partiamo da **0** e quindi scrivo **0**
1. aggiungo **1** cioe' **0+1 = 1** e quindi scrivo **1**
2. aggiungo **1** cioe' **1+1** siccome ottengo 2 e 2 non appartiene al sistema devo andare a capo cioe' prendo una coppia e 0 unita'
1+1 =10
cioe' ogni volta che ottengo 2 devo scrivere **0** e riportare un'unita' nello spazio prima dello **0**, guarda a destra dove ti ho indicato l'operazione in rosso ed il riporto in verde

1←
1 +
1 =
1 0

3. aggiungo **1** cioe' **10+1= 11** e quindi scrivo **11**
4. aggiungo **1** cioe' **11+1** allora sommando 1+1 ottengo **0** e riporto di **1** poi devo sommare l'**1** riportato ed ottengo **0** e riporto di **1** cioe' **11+1 = 100** : a destra, in rosso, la somma in colonna con in verde i riporti

1←1←
1 1 +
1 =
1 0 0

5. aggiungo **1** cioe' **100+1 = 101** ottengo **101**
6. aggiungo **1** cioe' **101+1** devo sommare 1+1 e quindi scrivo **0** e riporto di **1** sommo il riporto con lo **0** ed ottengo **110**
7. aggiungo **1** cioe' **110+1 = 111** ottengo **101**
8. aggiungo **1** cioe' **111+1** allora sommando 1+1 ottengo **0** e riporto di **1** poi devo sommare l'**1** riportato ed ottengo **0** e riporto di **1** poi devo sommare ancora l'**1** riportato ed ottengo **0** e riporto di **1** cioe' **111+1 = 1000** : a destra, in rosso, la somma in colonna con in verde il riporto

1←1←1←
1 1 1 +
1 =
1 0 0 0

9. aggiungo **1** cioe' **1000+1 = 1001** ottengo **1001**
10. aggiungo **1** cioe' **1001+1** devo sommare 1+1 e quindi scrivo **0** e riporto di **1** sommo il riporto con lo **0** ed ottengo **1010**

eccetera......

E' fondamentale conoscere perfettamente a memoria i valori dei vari numeri binari composti da un solo 1 e dagli zeri, cioe' delle potenze del 2 almeno fino a 1024: saranno le basi per il calcolo binario.
Per ricordartele meglio osserva che il numero degli zeri del numero binario corrisponde all'esponente del 2 nel sistema decimale:

$$1_{10} = 2^0{}_{10} = 1_2$$
$$2_{10} = 2^1{}_{10} = 10_2$$
$$4_{10} = 2^2{}_{10} = 100_2$$
$$8_{10} = 2^3{}_{10} = 1000_2$$
$$16_{10} = 2^4{}_{10} = 10000_2$$
$$32_{10} = 2^5{}_{10} = 100000_2$$
$$64_{10} = 2^6{}_{10} = 1000000_2$$
$$128_{10} = 2^7{}_{10} = 10000000_2$$
$$256_{10} = 2^8{}_{10} = 100000000_2$$
$$512_{10} = 2^9{}_{10} = 1000000000_2$$
$$1024_{10} = 2^{10}{}_{10} = 10000000000_2$$

In informatica, siccome 1024 e' vicino a 1000, si considera il prefisso kilo per il valore 1024, quindi avremo, ad esempio parlando di bytes:
1 Kilobyte = 1024 bytes
1 Megabyte = 1024 x 1024 = 1024^2 = 1048576 bytes
1 Gigabyte = 1024 x 1024 x 1024 = 1024^3 = 1073741824 bytes
1 Terabyte = 1024 x 1024 x 1024 x 1024 = 1024^4 = 1099511627776 bytes

---

d) Passaggio dal sistema decimale al sistema binario

Trasformare un numero dal sistema decimale al sistema binario e' un'operazione relativamente semplice: basta dividere il numero successivamente per 2 sinche' otteniamo 0 e tenere conto dei resti.
Tali resti, scritti in ordine inverso ci daranno il numero trasformato in binario.

Esempio:
Trasformare il numero **140_{10}** in numero binario .
Vicino al numero scrivo il resto mentre il quoziente lo scrivo sotto:

	quoziente	resto
140 diviso 2 da' **70** con resto di **0**	140	0
70 diviso 2 da' **35** con resto di **0**	70	0
35 diviso 2 da' **17** con resto di **1**	35	1
17 diviso 2 da' **8** con resto di **1**	17	1
8 diviso 2 da' **4** con resto di **0**	8	0
4 diviso 2 da' **2** con resto di **0**	4	0
2 diviso 2 da' **1** con resto di **0**	2	0
1 diviso 2 da' **0** con resto di **1**	1	1
	0	

Il numero binario corrispondente a 140_{10} e' 10001100_2
Ti ricordo che, come abbiamo gia' visto, la divisione equivale al raggruppare.

Per esercizio trasforma in binari i seguenti numeri decimali:
a) 46_{10} =
b) 159_{10} =
c) 678_{10} =
d) 1024_{10} =
e) 13282_{10} =
Svolgimento:
a) 46_{10} =
Io, qui come negli esercizi successivi, ti spiego tutti i passaggi, uno ad uno, pero' di solito si procede semplicemente con il mettere i numeri in colonna per trovarne i resti della divisione per 2
Vicino al numero scrivo il resto mentre il quoziente lo scrivo sotto il numero stesso

	quoziente	resto
46 diviso 2 da' **23** con resto di **0**	46	0
23 diviso 2 da' **11** con resto di **1**	23	1
11 diviso 2 da' **5** con resto di **1**	11	1
5 diviso 2 da' **2** con resto di **1**	5	1
2 diviso 2 da' **1** con resto di **0**	2	0
1 diviso 2 da' **0** con resto di **1**	1	1
	0	

Quindi: $46_{10} = 101110_2$

b) $159_{10} =$

Io, qui come negli esercizi successivi, ti spiego tutti i passaggi, uno ad uno, pero' di solito si procede semplicemente con il mettere i numeri in colonna per trovarne i resti della divisione per 2

Vicino al numero scrivo il resto mentre il quoziente lo scrivo sotto il numero stesso

	quoziente	resto
159 diviso 2 da' **79** con resto di **1**	159	1
79 diviso 2 da' **39** con resto di **1**	79	1
39 diviso 2 da' **19** con resto di **1**	39	1
19 diviso 2 da' **9** con resto di **1**	19	1
9 diviso 2 da' **4** con resto di **1**	9	1
4 diviso 2 da' **2** con resto di **0**	4	0
2 diviso 2 da' **1** con resto di **0**	2	0
1 diviso 2 da' **0** con resto di **1**	1	1
	0	

Quindi: $159_{10} = 10011111_2$

c) $678_{10} =$

Vicino al numero scrivo il resto mentre il quoziente lo scrivo sotto il numero stesso.

	quoziente	resto
678 diviso 2 da' **339** con resto di **0**	678	0
339 diviso 2 da' **169** con resto di **1**	339	1
169 diviso 2 da' **84** con resto di **1**	169	1
84 diviso 2 da' **42** con resto di **0**	84	0
42 diviso 2 da' **21** con resto di **0**	42	0
21 diviso 2 da' **10** con resto di **1**	21	1
10 diviso 2 da' **5** con resto di **0**	10	0
5 diviso 2 da' **2** con resto di **1**	5	1
2 diviso 2 da' **1** con resto di **0**	2	0
1 diviso 2 da' **0** con resto di **1**	1	1
	0	

Quindi: $678_{10} = 1010100110_2$

d) $1024_{10} =$

Vicino al numero scrivo il resto mentre il quoziente lo scrivo sotto il numero stesso

	quoziente	resto
1024 diviso 2 da' **512** con resto di **0**	1024	0
512 diviso 2 da' **256** con resto di **0**	512	0
256 diviso 2 da' **128** con resto di **0**	256	0
128 diviso 2 da' **64** con resto di **0**	128	0
64 diviso 2 da' **32** con resto di **0**	64	0
32 diviso 2 da' **16** con resto di **0**	32	0
16 diviso 2 da' **8** con resto di **0**	16	0
8 diviso 2 da' **4** con resto di **0**	8	0
4 diviso 2 da' **2** con resto di **0**	4	0
2 diviso 2 da' **1** con resto di **0**	2	0
1 diviso 2 da' **0** con resto di **1**	1	1
	0	

Quindi: $1024_{10} = 10000000000_2$

E' uno dei numeri "speciali" di cui abbiamo gia' parlato e precisamente 2^{10}

e) $13282_{10} =$

Vicino al numero scrivo il resto mentre il quoziente lo scrivo sotto il numero stesso

	quoziente	resto
13282 diviso 2 da' **6641** con resto di **0**	13282	0
6641 diviso 2 da' **3320** con resto di **1**	6641	1
3320 diviso 2 da' **1660** con resto di **0**	3320	0
1660 diviso 2 da' **830** con resto di **0**	1660	0
830 diviso 2 da' **415** con resto di **0**	830	0
415 diviso 2 da' **207** con resto di **1**	415	1
207 diviso 2 da' **103** con resto di **1**	207	1
103 diviso 2 da' **51** con resto di **1**	103	1
51 diviso 2 da' **25** con resto di **1**	51	1
25 diviso 2 da' **12** con resto di **1**	25	1
12 diviso 2 da' **6** con resto di **0**	12	0
6 diviso 2 da' **3** con resto di **0**	6	0
3 diviso 2 da' **1** con resto di **1**	3	1
1 diviso 2 da' **0** con resto di **1**	1	1
	0	

Quindi: $13282_{10} = 11001111100010_2$

e) Passaggio dal sistema binario al sistema decimale

Per passare dal sistema binario al sistema decimale utilizzeremo la forma polinomiale dei numeri binari, cioe' ogni numero binario puo' essere pensato in forma decimale come un polinomio a base 2; cioe' ad esempio:

$10011_2 = [1\cdot2^4 + 0\cdot2^3 + 0\cdot2^2 + 1\cdot2^1 + 1\cdot2^0]_{10} = [16+2+1]_{10} = 19_{10}$

cioe' la cifra binaria (0 od 1) va moltiplicato per la potenza del due corrispondente al posto che tale cifra occupa nel numero binario stesso.

Cifra binaria	......	undicesima	decima	nona	ottava	settima	sesta	quinta.	quarta	terza	seconda	prima
Potenza del 2	......	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Valore potenza	......	1024	512	256	128	64	32	16	8	4	2	1

Notare che l'esponente del 2 e' sempre inferiore di 1 del posto della cifra perche' si parte da zero: cifra ottava potenza 7, cifra sesta potenza 5,...
Ad esempio, se hai il numero **10011011** composto di 8 cifre mettilo mentalmente in tabella ed avrai :

Cifra binaria	......	undicesima	decima	nona	ottava	settima	sesta	quinta.	quarta	terza	seconda	prima
Potenza del 2	......	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Valore potenza	......	1024	512	256	128	64	32	16	8	4	2	1
Numero dato	......				1	0	0	1	1	0	1	1

Adesso somma i numeri corrispondenti alle cifre sopra gli **1**:
128+16+8+2+1 = 155
cioe':
$10011011_2 = 155_{10}$
Comunque, di solito, negli esercizi, senza passare per la forma polinomiale, e' preferibile ricordare la successione:
1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096,...
delle potenze del 2: per ricordartela osserva che ogni numero e' doppio del precedente.
Essendo il numero binario composto di 1 e di 0, basta associare ad ogni numero 1 il valore del posto che occupa.
Faccio un esempio.
Trasformare il numero binario **110011101_{10}** in numero decimale; scrivo, sopra ogni numero 1 il valore corrispondente, naturalmente cominciando dall'1 piu' a destra e procedendo verso sinistra:

256 128 16 8 4 1

1 1 0 0 1 1 1 0 1

Scrivo 1 sopra il primo 1 a destra, poi sopra lo 0 dovrei scrivere 2, ma essendo 0 lo salto; poi scrivo 4 sopra l'1 al terzo posto; 8 sopra l'uno al quarto posto e 16 sopra l'1 al quinto posto; al sesto posto dovrei scrivere 32 ma siccome c'e' lo zero lo salto e cosi anche al settimo posto dovrei scrivere 64 ma non lo scrivo perche' c'e' lo zero, scrivo invece 128 sopra l'1 all'ottavo posto e 256 sopra l'1 al nono posto.

Quindi:

$110011101_2 = 256 + 128 + 16 + 8 + 4 + 1 = 413_{10}$

Per esercizio trasforma in decimali i seguenti numeri binari:

a) $111001_2 =$

Ricordando la successione delle potenze del due

1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096,...

scrivo, sopra ogni numero 1 il valore corrispondente, naturalmente cominciando dall'1 piu' a destra e procedendo verso sinistra

32 16 8 1

1 1 1 0 0 1

Scrivo 1 sopra il primo 1 a destra, poi sopra lo 0 dovrei scrivere 2, ma essendo 0 lo salto, poi dovrei scrivere 4 sopra la cifra al terzo posto,ma essendo 0 non scrivo niente, scrivo 8 sopra l'uno al quarto posto e 16 sopra l'1 al quinto posto, e 32 sopra l'1 al sesto posto.

Quindi:

$111001_2 = 32 + 16 + 8 + 1 = 57_{10}$

b) $1010101010_2 =$ Svolgimento

Ricordando la successione delle potenze del due:

1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096,...

scrivo, sopra ogni numero 1 il valore corrispondente, naturalmente cominciando dall'1 piu' a destra e procedendo verso sinistra:

512 128 32 8 2

1 0 1 0 1 0 1 0 1 0

Dovrei scrivere 1 sopra la prima cifra a destra, ma essendo 0 lo salto, poi sopra l'1 al secondo posto scrivo 2, poi dovrei scrivere 4 sopra la cifra al terzo posto,ma essendo 0 non scrivo niente, scrivo 8 sopra l'uno al quarto posto e poi dovrei scrivere 16 sopra la cifra al quinto posto, ma essendo 0 non scrivo niente, scrivo 32 sopra l'uno al sesto posto e poi dovrei scrivere 64 sopra la cifra al settimo posto, ma essendo 0 non scrivo niente, scrivo 128 sopra l'uno all'ottavo posto e poi dovrei scrivere 252 sopra la cifra al nono posto, ma essendo 0 non scrivo niente, ed infine scivo 512 sopra l'1 al decimo posto.

Quindi

$1010101010_2 = 512 + 128 + 32 + 8 + 2 = 682_{10}$

c) $11111111_2 =$ Svolgimento

Ricordando la successione delle potenze del due

1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096,...

scrivo, sopra ogni numero 1 il valore corrispondente, naturalmente cominciando dall'1 piu' a destra e procedendo verso sinistra .

128 63 32 16 8 4 2 1

1 1 1 1 1 1 1 1

Scrivo 1 sopra la prima cifra a destra , poi sopra l'1 al secondo posto scrivo 2, poi scrivo 4 sopra l'1 al terzo posto, scrivo 8 sopra l'uno al quarto posto, scrivo 16 sopra l'1 al quinto posto, scrivo 32 sopra l'uno al sesto posto, scrivo 64 sopra l'1 al settimo posto, ed infine scrivo 128 sopra l'uno all'ottavo posto.

Quindi:

$$\mathbf{11111111_2 = 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255_{10}}$$

Qui di seguito metto una tabella con corrispondenti i primi 20 numeri binari ed i numeri decimali, potrebbe esserti utile.

Binario	Binario in forma polinomiale	Decimale
0	$0 \cdot 2^0 = 0 \cdot 1 = 0$	0
1	$1 \cdot 2^0 = 1 \cdot 1 = 1$	1
10	$1 \cdot 2^1 + 0 \cdot 2^0 = 2+0 = 2$	2
11	$1 \cdot 2^1 + 1 \cdot 2^0 = 2+1 = 3$	3
100	$1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 4+0+0 = 4$	4
101	$1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 4+0+1 = 5$	5
110	$1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 4+2+0 = 6$	6
111	$1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 4+2+1 = 7$	7
1000	$1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 8+0+0+0 = 8$	8
1001	$1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 8+0+0+1 = 9$	9
1010	$1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 8+0+2+0 =10$	10
1011	$1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 8+0+1+1 =11$	11
1100	$1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 8+4+0+0 =12$	12
1101	$1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 8+4+0+1 =13$	13
1110	$1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 8+4+2+0 =14$	14
1111	$1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 8+4+2+1 =15$	15
10000	$1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 16+0+0+0+0 =16$	16
10001	$1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 16+0+0+0+1 =17$	17
10010	$1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 16+0+0+2+0 =18$	18
10011	$1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 16+0+0+2+1 =19$	19
10100	$1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 16+0+4+0+0 =20$	20
....	....	....

f) Le operazioni nel sistema binario

Veramente, per informatica, sarebbe sufficiente fare solamente la somma; la differenza si puo' ridurre alla somma complementare; come vedremo, il prodotto e' una somma ripetuta ed il quoziente e' una differenza ripetuta. Quindi in informatica bastera' costruire un solo circuito di base che sia capace di fare la somma fra due numeri binari elementari (0 ed 1) e le altre operazioni si baseranno sempre sulla somma e quindi sullo stesso circuito;

Pero' matematicamente, mi sembra giusto considerare sui numeri binari tutte e quattro le operazioni e quindi, anche se non utile per informatica, vediamo di studiarle tutte.

- somma
- la differenza come somma complementare
- differenza
- prodotto
- quoziente

1) Somma fra numeri binari

Ti ripeto che, come abbiamo gia' detto, La tabella per eseguire le operazioni di somma (come la tavola pitagorica per la somma) e' la seguente

+	0	1
0	0	1
1	1	10

Cioe'

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 10

Nel sistema binario, quando eseguo la somma 1+1 scrivo 0 e riporto 1 nella colonna delle coppie **1+1=10**: cioe' per le operazioni, la coppia corrisponde alla decina nel sistema decimale.
A fianco la somma 1+1=10 col riporto in verde per il sistema binario

1←
1 +
1 =
10

Vediamo su un esempio, come si esegue una somma fra numeri binari
sommare: **1100010101** e **100010111** prima li metto in colonna (si parte sempre da destra):

1	1	0	0	0	1	0	1	0	1	+
	1	0	0	0	1	0	1	1	1	

Adesso sommo partendo da destra: sopra, in verde e carattere piu' piccolo ti scrivo i riporti

1←	1←			1←		1←	1←	1←			
	1	1	0	0	0	1	0	1	0	1	+
		1	0	0	0	1	0	1	1	1	
1	0	0	0	0	1	0	1	1	0	0	

Esercizi
Esegui le seguenti somme fra numeri binari:
a) 10010101101 + 1110000 = Svolgimento
Prima li metto in colonna (si parte sempre da destra):

1	0	0	1	0	1	0	1	1	0	1	+
				1	1	1	0	0	0	0	

Adesso sommo partendo da destra; sopra, in verde e carattere piu' piccolo ti scrivo i riporti

		1←		1←		1←					
1	0	0	1	0	1	0	1	1	0	1	+
				1	1	1	0	0	0	0	
1	0	1	0	0	0	1	1	1	0	1	

b) 1111111111+100000001 = Svolgimento
prima li metto in colonna (si parte sempre da destra)

1	1	1	1	1	1	1	1	1	1	+
	1	0	0	0	0	0	0	0	1	

Adesso sommo partendo da destra: sopra, in verde e carattere piu' piccolo ti scrivo i riporti

1←	1←	1←	1←	1←	1←	1←	1←	1←	1←		
	1	1	1	1	1	1	1	1	1	1	+
		1	0	0	0	0	0	0	0	1	
1	0	1	0	0	0	0	0	0	0	0	

Trasforma i seguenti numeri in binari, esegui le operazioni e poi riporta il risultato in forma decimale
a) $67_{10} + 35_{10}$ = Svolgimento
Questo e' un esercizio abbastanza "completo", che ci permette di ripassare quanto fatto in precedenza. Io lo svolgo facendo tutti i passaggi, pero' tu, una volta capito il metodo, dovresti eseguire gli esercizi il piu' velocemente possibile.

Prima trasformo 67_{10} in numero binario
Vicino al numero scrivo il resto mentre il quoziente lo scrivo sotto il numero stesso

	quoziente	resto
67 diviso 2 da' **33** con resto di **1**	67	1
33 diviso 2 da' **16** con resto di **1**	33	1
16 diviso 2 da' **8** con resto di **0**	16	0
8 diviso 2 da' **4** con resto di **0**	8	0
4 diviso 2 da' **2** con resto di **0**	4	0
2 diviso 2 da' **1** con resto di **0**	2	0
1 diviso 2 da' **0** con resto di **1**	1	1
	0	

Quindi:
67_{10} = 10000112

Ora trasformo **35_{10}** in numero binario:

35 diviso 2 da' **17** con resto di **1**
17 diviso 2 da' **8** con resto di **1**
8 diviso 2 da' **4** con resto di **0**
4 diviso 2 da' **2** con resto di **0**
2 diviso 2 da' **1** con resto di **0**
1 diviso 2 da' **0** con resto di **1**

quoziente	resto
35	1
17	1
8	0
4	0
2	0
1	1
0	

Quindi:
35_{10} = 1000112

Adesso li metto in colonna e sommo partendo da destra: sopra, in verde e carattere piu' piccolo ti scrivo i riporti

				1←	1←		
1	0	0	0	0	1	1	+
	1	0	0	0	1	1	
1	1	0	0	1	1	0	

Quindi:
10000112 + 1000112 = 11001102

Adesso trasformo il risultato in forma decimale

Ricordando la successione delle potenze del due:
1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096,...
scrivo, sopra ogni numero 1 il valore corrispondente, naturalmente cominciando dall'1 piu' a destra e procedendo verso sinistra.

64	32			4	2	
1	1	0	0	1	1	0

Sopra lo 0 non scrivo niente poi scrivo 2 sopra il primo 1 a destra (seconda cifra), poi sopra il secondo 1 (terza cifra) scrivo 4, poi dovrei scrivere 8 sopra la cifra al quarto posto,ma essendo 0 non scrivo niente, poi dovrei scrivere 16 sopra la cifra al quinto posto,ma essendo 0 non scrivo niente, scrivo 32 sopra l'uno al sesto posto e 64 sopra l'1 al settimo posto.
Quindi:
$1100110_2 = 64 + 32 + 4 + 2 = 102_{10}$

b) **$680_{10} + 378_{10} =$** Svolgimento
Io lo svolgo facendo tutti i passaggi, pero' tu, una volta capito il metodo, dovresti eseguire gli esercizi il piu' velocemente possibile.

Prima trasformo **680_{10}** in numero binario
Vicino al numero scrivo il resto mentre il quoziente lo scrivo sotto il numero stesso

	quoziente	resto
680 diviso 2 da' **340** con resto di **0**	680	0
340 diviso 2 da' **170** con resto di **0**	340	0
170 diviso 2 da' **85** con resto di **0**	170	0
85 diviso 2 da' **42** con resto di **1**	85	1
42 diviso 2 da' **21** con resto di **0**	42	0
21 diviso 2 da' **10** con resto di **1**	21	1
10 diviso 2 da' **5** con resto di **0**	10	0
5 diviso 2 da' **2** con resto di **1**	5	1
2 diviso 2 da' **1** con resto di **0**	2	0
1 diviso 2 da' **0** con resto di **1**	1	1
	0	

Quindi:
680_{10} = 10101010002

Ora trasformo **378_{10}** in numero binario

	quoziente	resto
378 diviso 2 da' **189** con resto di **0**	378	1
189 diviso 2 da' **94** con resto di **1**	189	1
94 diviso 2 da' **47** con resto di **0**	94	0
47 diviso 2 da' **23** con resto di **1**	47	1
23 diviso 2 da' **11** con resto di **1**	23	1
11 diviso 2 da' **5** con resto di **1**	11	1
5 diviso 2 da' **2** con resto di **1**	5	1
2 diviso 2 da' **1** con resto di **0**	2	0
1 diviso 2 da' **0** con resto di **1**	1	1
	0	

Quindi:
378_{10} = 1011110102

Adesso li metto in colonna e sommo partendo da destra: sopra, in verde e carattere piu' piccolo ti scrivo i riporti

1←	1←	1←	1←	1←	1←	1←					
	1	0	1	0	1	0	1	0	0	0	+
		1	0	1	1	1	1	0	1	0	
1	0	0	0	0	1	0	0	0	1	0	

Quindi:
10101010002 + 1011110112 = 100001000102

Adesso trasformo il risultato in forma decimale

Ricordando la successione delle potenze del due
1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096,...

scrivo, sopra ogni numero 1 il valore corrispondente, naturalmente cominciando dall'1 piu' a destra e procedendo verso sinistra

---

1024					32				2	0
1	0	0	0	0	1	0	0	0	1	0

Non scrivo niente sopra la prima cifra a destra perche' e' zero, poi scrivo 2 sopra la seconda cifra a destra (perche' e' 1), poi sopra la terza cifra, essendo 0, dovrei scrivere 4 ma non scrivo niente, lo stesso sopra la quarta (8) e la quinta cifra (16) che sono zeri; poi scrivo 32 sopra l'1 al sesto posto, poi dovrei scrivere 64 sopra la cifra al settimo posto,ma essendo 0 non scrivo niente, lo stesso sopra l'ottava (128), la nona (256) e la decima cifra(512) che sono zeri non scrivo niente scrivo invece 1024 sopra l'uno all'undicesimo posto.
Quindi
$\mathbf{10000100011_2 = 1024 + 32 + 2 + 0 = 1058_{10}}$
Quindi:
$\mathbf{680_{10} + 378_{10} = 1058_{10}}$

Domanda del solito Pierino:
"Ma non avremmo fatto prima a sommare normalmente per trovare il risultato?"
Certamente, noi avremmo fatto prima facendo i calcoli col sistema decimale, pero' in questo modo possiamo far fare tutto il lavoro ad una macchina che ci restituira' il risultato e noi ci limiteremo a scrivere i dati e leggere il risultato

2) La differenza come somma complementare

Vediamo come e' possibile, con l'uso dei complementari, trasformare l'operazione di sottrazione in un'operazione di somma, senza quindi dover ricorrere a prestiti dalle colonne precedenti.

Consideriamo di avere due numeri **a** e **b** ad esempio di 4 cifre con **a<b** e consideriamo la sottrazione **a-b.**
Posso scrivere:
$\mathbf{a-b = a - b + (9999 + 1 -10000) =}$
cioe' aggiungo e tolgo 10000 e quindi il valore non varia: infatti 9999+1 =10000
Adesso faccio cadere le parentesi
$\mathbf{= a - b + 9999 + 1 -10000 =}$
posso anche scrivere
$\mathbf{= a + (9999 - b + 1) -10000 =}$
evidenzio l'operazione di complementare
$\mathbf{= a + [(9999 - b) + 1] -10000 =}$
il termine **(999-b)** e' il complementare di **b**, inoltre il numero 10000 non interviene nel risultato che per l'1 che sta davanti al risultato, quindi potremmo eliminarlo tranquillamente e la mia operazione da differenza e' diventata una somma. Questo metodo si chiama *end-around carry*.

Esempio:
Eseguire la differenza **8765 - 3210 =**
Scrivo il complemento a 9 di 3210: basta mettere al posto di ogni cifra quello che manca per arrivare a 9 quindi ottengo:

complemento a 9 di 3210 = 6789
Scrivo in colonna:

8765 +
6789 =
1 5554

Adesso tolgo l'1 da davanti (equivale a fare -10000) e lo aggiungo alla cifra delle unita' ed ottengo il risultato

8765 +
6789 =

1 5554
→ 1

5555

Come abbiamo fatto con la sottrazione decimale, possiamo fare con la sottrazione binaria; eseguire la differenza binaria **110010101 - 10110110** =
Scrivo il complemento a 2 di 10110110; basta mettere al posto di ogni cifra quello che manca per arrivare a 1; quindi ottengo:
complemento a 2 di 10110110 = 01001001
Lascio lo 0 iniziale; per mostrartelo meglio scrivo in colonna:

110010101 +
01001001 =
1 10011110

Adesso tolgo l'1 da davanti e lo aggiungo alla cifra delle unita' ed ottengo il risultato:

110010101 +
01001001 =

1 11011110
→ 1

11011111

Se trasformi in decimale si tratta della sottrazione **405-182 = 223**.
Questo e' il modo in cui funziona la tua calcolatrice tascabile quando fai una sottrazione

3) Differenza fra numeri binari

Per eseguire le operazioni di differenza seguiremo queste regole:

0 - 0 = 0
1 - 0 = 1
1 - 1 = 0

Nel sistema binario, quando eseguo la differenza **10-1** per eseguire **0-1** devo andare "in prestito" dalla cifra precedente e quindi valendo l'1 della seconda cifra **2=1+1** devo fare **(1+1)-1 = 1**, di conseguenza al posto dell'**1** della seconda cifra avro' **0.**
A fianco la differenza per il sistema binario **10-1= 01** col il "prestito" indicato in verde

1→	1+1	
1	0	-
	1	=
0	1	

Vediamo su un esempio; come si esegue una differenza fra numeri binari.
Sottrarre: **100010111** da **1100110101;** prima li metto in colonna (si parte sempre da destra):

1	1	0	0	1	1	0	1	0	1	-
	1	0	0	0	1	0	1	1	1	

Adesso sottraggo partendo da destra; sopra, in verde e carattere piu' piccolo ti scrivo i "prestiti"

					→1+1	→1	→1+1	→1+1		
1	1	0	0	1	1	0	1	0	1	-
	1	0	0	0	1	0	1	1	1	
1	0	0	0	0	1	1	1	1	0	

Esercizi

Esegui le seguenti differenze fra numeri binari:
a) 10100100101 - 1101001010 = Svolgimento
Prima li metto in colonna (si parte sempre da destra):

1	0	1	0	0	1	0	0	1	0	1	-
	1	1	0	1	0	0	1	0	1	0	

Adesso sottraggo partendo da destra; sopra, in verde e carattere piu' piccolo ti scrivo i prestiti

	→1	→1+1		→1+1					→1+1		
1	0	0	1	0	1	0	1	1	0	1	-
	1	1	0	1	0	0	1	0	1	0	
-	-	1	0	1	1	0	0	0	1	1	

b) 10000000000 - 1011111111 **=** Svolgimento

Prima li metto in colonna (si parte sempre da destra)

1	0	0	0	0	0	0	0	0	0	0	-
	1	0	1	1	1	1	1	1	1	1	

Adesso sottraggo partendo da destra; sopra, in verde e carattere piu' piccolo ti scrivo i prestiti

	→1	→1	→1	→1	→1	→1	→1	→1	→1	→1+1	
1	0	0	0	0	0	0	0	0	0	0	-
	1	0	1	1	1	1	1	1	1	1	
-	-	1	0	0	0	0	0	0	0	1	

4) Prodotto fra numeri binari

La tabella per eseguire le operazioni di prodotto e' la seguente

·	0	1
0	0	0
1	0	1

Cioe'

$0 \cdot 0 = 0$

$0 \cdot 1 = 0$

$1 \cdot 0 = 0$

$1 \cdot 1 = 1$

Lo **0** e' detto anche "elemento assorbente" perche' moltiplicato per qualunque numero lo "assorbe" facendolo diventare uguale a se' stesso

numero ·0 = 0·numero = 0

Vediamo su un semplice esempio; come si esegue un prodotto fra numeri binari.
Moltiplicare: 1100101 e 1001. Prima notiamo che, se moltiplichiamo il numero sopra per **1,** otteniamo sempre il numero di sopra quindi basta riportare il numero sopra per ogni cifra **1** (opportunamente posizionato) e una fila di zeri per ogni cifra **0**

			1	1	0	0	1	0	1	·
						1	0	0	1	
			1	1	0	0	1	0	1	
		0	0	0	0	0	0	0	-	
	0	0	0	0	0	0	0	-	-	
1	1	0	0	1	0	1	-	-	-	

o, piu' semplicemente, saltiamo le file di zeri e consideriamo solamente i termini effettivi poi sommiamo le due righe:

			1	1	0	0	1	0	1	·
						1	0	0	1	
			1	1	0	0	1	0	1	
1	1	0	0	1	0	1	-	-	-	
1	1	1	0	0	0	1	1	0	1	

Nel caso preso in esame abbiamo solo due 1 per il moltiplicatore (1001); se invece gli 1 sono 3, 4, 5, e' piuttosto difficile eseguire tutta assieme la somma delle righe ottenute, quindi conviene sommare a parte la prima con la seconda, il risultato con la terza, il risultato con la quarta eccetera

Vediamo un esempio con 3 unita'.

Moltiplicare: **1100111** e **10110**

				1	1	0	0	1	1	1	·
						1	0	1	1	0	
			1	1	0	0	1	1	1	-	
		1	1	0	0	1	1	1	-	-	
	1	0	0	1	1	0	1	0	1	0	somma parziale
	1	1	0	0	1	1	1	-	-	-	
1	0	1	1	0	1	0	0	0	1	0	

Esercizi

esegui i seguenti prodotti fra numeri binari:

a) 101001·1010 = Svolgimento

Avendo solamente due **1** nel moltiplicatore procedo direttamente al calcolo

			1	0	1	0	0	1	·
					1	0	1	0	
		1	0	1	0	0	1	-	
1	0	1	0	0	1	-	-	-	
1	1	0	0	1	1	0	1	0	

b) 10010 ·1001100= Svolgimento

Notiamo che, avendo il moltiplicando solamente due cifre **1**, mentre il moltiplicatore ne ha tre, ci conviene, usando la proprieta' commutativa della moltiplicazione, scrivere in questo modo:

1001100 ·10010 =

Avendo ora solamente due **1** nel moltiplicatore, procedo direttamente al calcolo : proprieta' commutativa

				1	0	0	1	1	0	0	·
						1	0	0	1	0	
		1	0	0	1	1	0	0	-		
1	0	0	1	1	0	0	-	-	-	-	
1	0	1	0	1	0	1	1	0	0	0	

Quindi:

10010 ·1001100 = 10101011000

c) 101010 ·1001100= Svolgimento

procedo direttamente al calcolo eseguendo le somme parziali nello svolgimento stesso

					1	0	1	0	1	0	·	
				1	0	0	1	1	0	0		
			1	0	1	0	1	0	-	-		
		1	0	1	0	1	0	-	-	-		
		1	1	1	1	1	1	0	0	0		prima somma parziale
1	0	1	0	1	0	-	-	-	-	-		
1	1	1	0	0	1	1	1	0	0	0		seconda somma parziale e risultato

Quindi :

101010 ·1001100 = 11100111000

d) 1010110 ·1001110= Svolgimento

procedo direttamente al calcolo eseguendo le somme parziali nello svolgimento stesso

						1	0	1	0	1	1	0	·	
						1	0	0	1	1	1	0		
					1	0	1	0	1	1	0	-		
				1	0	1	0	1	1	0	-	-		
			1	0	0	0	0	0	0	1	0	0		prima somma parziale
			1	0	1	0	1	1	0	-	-	-		
		1	0	0	1	0	1	1	0	1	0	0		seconda somma parziale
1	0	1	0	1	1	0	-	-	-	-	-	-		
1	1	0	1	0	0	0	1	1	0	1	0	0		terza somma parziale e risultato

Quindi:

1010110 ·1001110 = 1101000110100

5) Quoziente fra numeri binari

In pratica eseguire la divisione fra numeri binari sara' equivalente ad eseguire ripetute sottrazioni.
Vediamo, su un semplice esempio, come si esegue una divisione fra numeri binari.
Eseguire la divisione fra **1011010** e **1001**:

```
1011010 | 1001
        |------
1001      1010
----
- - 1001
- - 1001
--------
  - - - - 0
```

Ti spiego passo passo come si procede; le cifre del divisore sono 4; quindi considero 4 cifre del dividendo, controllo che le 4 cifre del dividendo diano un numero di valore uguale o superiore alle quattro cifre del divisore; in questo caso e' vero, se invece fosse di valore inferiore, devo prendere una cifra in piu' nel dividendo.
Come vedere subito se e' superiore od uguale
1001 in 1011 ci sta, quindi scrivo 1 nel quoziente e scrivo 1001 sotto le cifre del dividendo; quindi eseguo la sottrazione: ottengo 10.
Aggiungo una cifra del dividendo ed ottengo 100, 1001 in 100 non ci sta; quindi scrivo 0 nel quoziente (la cifra rossa) e abbasso un'altra cifra, ottengo 1001.
1001 in 1001 ci sta una volta, scrivo 1 nel quoziente e scrivo 1001 sotto 1001; eseguo la sottrazione ed ottengo 0.
Siccome ho ancora una cifra, lo 0, nel divisore lo abbasso . 1001 in 0 non ci sta, quindi scrivo 0 come ultima cifra del quoziente (quella fucsia). Quindi ho ottenuto:
1011010 : 1001 = 1010

Esercizi

esegui le seguenti divisioni fra numeri binari:
a) 111000 : 1000 = Svolgimento

```
111000 | 1000
       |------
1000     111
----
- 1100
- 1000
------
 - 1000
   1000
 ------
    - - - -
```

Ti spiego passo passo come si procede:
le cifre del divisore (1000) sono 4 quindi considero 4 cifre del dividendo (1110): vedo che le 4 cifre del divisore danno un numero di valore inferiore al valore delle quattro cifre del divisore, cioe' 1000 in 1110 ci sta, quindi scrivo 1 nel quoziente e scrivo 1000 sotto le cifre del dividendo, ed eseguo la sottrazione: ottengo 110.
Aggiungo una cifra del dividendo ed ottengo 1100; 1000 in 1100 ci sta, quindi scrivo 1 nel quoziente e scrivo 1000 sotto le cifre del dividendo, quindi eseguo la sottrazione: ottengo 100; e abbasso l'ultima cifra, ottengo 1000.
1000 in 1000 ci sta una volta, scrivo 1 nel quoziente e scrivo 1000 sotto 1000; eseguo la sottrazione ed ottengo 0 (resto zero, divisione esatta); invece di 0 ho messo un insieme di trattini.
Quindi ho ottenuto:
111000 : 1000 = 111

b) 100010001 : 11011 = Svolgimento

100010001	**11011**
11011	**1010**
- - - 11100	
11011	
- - - - 11	

Ti spiego passo passo come si procede:
le cifre del divisore (11011) sono 5 quindi considero 5 cifre del dividendo (10001): vedo che le 5 cifre del divisore danno un numero di valore superiore al valore delle cinque cifre del dividendo, cioe' 11011 in 10001 non ci sta, quindi considero anche la sesta cifra del dividendo ed ottengo 100010.
Ora 11011 in 100010 ci sta, quindi scrivo 1 nel quoziente e scrivo 11011 sotto le cifre considerate del dividendo (partendo da destra), ed eseguo la sottrazione; ottengo 111

Aggiungo una cifra del dividendo ed ottengo 1110;
11011 in 1110 non ci sta, quindi scrivo 0 nel quoziente ed abbasso un'altra cifra; ottengo 11100.

11011 in 11100 ci sta, quindi scrivo 1 nel divisore e scrivo 11011 sotto le cifre considerate (partendo da destra), ed eseguo la sottrazione: ottengo 1.

e abbasso l'ultima cifra, ottengo 11.
11011 in 11 non ci sta una volta, scrivo 0 nel quoziente ed hon terminato le cifre, quindi 11 e' il resto

Ho ottenuto:
100010001 : 11011 da' 1010 con resto 11

Trasformando in decimale e' **273 : 27 da' 9 con resto di 3**

c) **100111111 : 111010 =** Svolgimento

```
100111111       | 111010
                +--------
 111101           101
----------
--1010111
   111010
----------
      --11101
```

Ti spiego passo passo come si procede:
le cifre del divisore (111010) sono 6 quindi considero 6 cifre del dividendo (100111): vedo che le 6 cifre del divisore danno un numero di valore superiore al valore delle sei cifre del dividendo, cioe' 111010 in 100111 non ci sta, quindi considero anche la settima cifra del dividendo ed ottengo 1001111.
Ora 111010 in 1001111 ci sta, quindi scrivo 1 nel quoziente e scrivo 111010 sotto le cifre considerate del dividendo (partendo da destra), ed eseguo la sottrazione: ottengo 10101.

Abbasso un altra cifra del dividendo ed ottengo 101011.
111010 in 101011 non ci sta, scrivo 0 nel quoziente e aggiungo anche l'ultima cifra del dividendo ottenendo 1010111.

Ora 111010 in 1010111 ci sta, quindi scrivo 1 nel quoziente e scrivo 111010 sotto le cifre considerate del dividendo (partendo da destra), ed eseguo la sottrazione: ottengo 11101 (che e' il resto).

Ho ottenuto:
100111111 : 111010 da' 101 con resto 11101

Trasformando in decimale e' **319 : 58 da' 5 con resto di 29**

g) I numeri con la virgola nel sistema binario

Anche nel sistema binario, come nel sistema decimale, e' possibile, quando otteniamo un resto nell'operazione di divisione, procedere ancora trovando le cifre decimali del numero; vediamo qui un esempio di come si procede.

Prima consideriamo il caso di un numero decimale finito, ad esempio partiamo dalla divisione decimale:
50 : 8 = 6,25 e vediamo come comportarci

110010 : 1000 =

```
110010,00 | 1000
          |-------
1000        110,01
-------
- 1001
  1000
  -------
 - - -10,00
      10 00
  -------
      -- --
```

Ti spiego passo passo come si procede: le cifre del divisore (1000) sono 4 quindi considero 4 cifre del dividendo (1100); 1000 in 1100 ci sta, quindi scrivo 1 nel quoziente e scrivo 1000 sotto le cifre considerate del dividendo, ed eseguo la sottrazione: ottengo 100.

Abbasso un altra cifra del dividendo ed ottengo 1001.
1000 in 1001 ci sta, quindi scrivo 1 nel quoziente e scrivo 1000 sotto le cifre considerate del dividendo (partendo da destra), ed eseguo la sottrazione: ottengo 1.

Abbasso l'ultima cifra del dividendo ed ottengo 10.
1000 in 10 non ci sta, scrivo 0 nel quoziente e considero un primo 0 dopo la virgola, ottengo 100.

1000 in 100 non ci sta, scrivo la virgola e lo zero (.0) nel quoziente e considero un secondo 0 dopo la virgola, ottengo 1000.

1000 in 1000 ci sta, quindi scrivo 1 nel quoziente e scrivo 1000 sotto le cifre considerate del dividendo , ed eseguo la sottrazione: ottengo 0.

Ho ottenuto:
110010 : 1000 = 110,01

Nota: per ritrasformare da binario a decimale le cifre dopo la virgola dobbiamo far riferimento alle potenze di ½, o meglio alle potenze negative di 2
$2^{-1} = (½)^1 = 1/2 = 0,5$ $2^{-2} = (½)^2 = 1/4 = 0,25$ $2^{-3} = (½)^3 = 1/8 = 0,125$ ......

Cifra binaria	......	terza	seconda	prima	virgola	prima	seconda	terza	quarta	quinta
Potenza del 2	......	2^2	2^1	2^0	,	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}
Valore potenza	......	4	2	1	,	0,5	0,25	0,125	0,0625	0,03125

Ad esempio, se hai il numero **101,101** mettilo mentalmente in tabella ed avrai

Cifra binaria	......	terza	seconda	prima	virgola	prima	seconda	terza	quarta	quinta
Potenza del 2	......	2^2	2^1	2^0	,	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}
Valore potenza	......	4	2	1	,	0,5	0,25	0,125	0,0625	0,03125
Numero dato	......	1	0	1	,	1	0	1		

Quindi il numero preso ad esempio, in decimali vale:
$4 + 0 + 1 + 0{,}5 + 0 + 0{,}125 = 5{,}625$
In pratica significa che ogni numero decimale inferiore ad 1 puo' essere rappresentato come somma di termini della successione geometrica di ragione 1/2. Sarebbe a dire che esiste una corrispondenza biunivoca fra i numeri razionali inferiori ad 1 e le ridotte della serie geometrica di ragione 1/2. E' anche una verifica del fatto che la serie geometrica di ragione 1/2 converge ad 1.
Non so a voi, ma a me sembra questo fatto sembra interessante e credo che meriterebbe un approfondimento.

Il nostro risultato 110,01 trasformato in decimali vale:
$4 + 2 + 0 + 0 + 0{,}25 = 6{,}25$

Consideriamo ora il caso di un numero decimale illimitato periodico; ad esempio, partiamo dalla divisione decimale:
$20 : 6 = 3{,}3333....$ e vediamo come comportarci:

$10100 : 110 =$

```
10100, 0000 | 110
            |------------
 110          11, 0101.....
-----
- 1000
   110
  -----
   10, 00
    1  10
  -----
     -  1000
         110
     --------
       - -  10
 eccetera...
```

Ti spiego passo passo come si procede:
le cifre del divisore (110) sono 3, quindi considero 3 cifre del dividendo (101)
110 in 101 non ci sta, quindi considero 4 cifre del dividendo: 1010.
Ora 110 in 1010 ci sta quindi scrivo 1 nel quoziente e scrivo 110 sotto le cifre considerate del dividendo (partendo da destra), ed eseguo la sottrazione: ottengo 100.

Abbasso un altra cifra del dividendo ed ottengo 1000.
110 in 1000 ci sta, quindi scrivo 1 nel quoziente e scrivo 110 sotto le cifre considerate del dividendo (partendo da destra), ed eseguo la sottrazione: ottengo 10 .

Considero un primo 0 dopo la virgola, ottengo 100.

110 in 100 non ci sta, scrivo la virgola e lo zero (.0) nel quoziente e considero un secondo 0 dopo la virgola, ottengo 1000.

110 in 1000 ci sta, quindi scrivo 1 nel quoziente e scrivo 110 sotto le cifre considerate del dividendo, ed eseguo la sottrazione: ottengo 10.

Considero un terzo 0 dopo la virgola, ottengo 100.

110 in 100 non ci sta, scrivo 0 nel quoziente e considero un quarto 0 dopo la virgola, ottengo 1000.

110 in 1000 ci sta, quindi scrivo 1 nel quoziente e scrivo 110 sotto le cifre considerate del dividendo, ed eseguo la sottrazione: ottengo 10.

Siccome ogni volta, eseguendo la sottrazione ottengo 10 avro' che le cifre dopo la virgola 01, si ripeteranno infinitamente e sono il periodo, cioe' il numero e' binario periodico di periodo 01 .

Ho ottenuto:
10100 : 110 = 11,0101010101........

7. bit e Byte

Vediamo ora, prima di passare ad altri sistemi di numerazione di parlare degli oggetti che saranno poi in informatica legati a tali sistemi

- bit
- Byte
- codice ascii

a) bit

La parola **bit** e' l'acronimo di **BInary digIT**, cioe' cifra binaria ed e' la minima quantita' di informazione: si riduce sempre ad uno **0** oppure ad un **1.**
Intuitivamente, siccome nel computer si ha una corrente che viene mandata ad impulsi regolari (ciclo di clock), tale corrente puo' passare (1) oppure puo' non passare (0) attraverso un circuito/interruttore ed allora, tramite il circuito stesso, l'informazione puo' essere spostata ad altri interruttori facendoli o meno aprire in modo che l'informazione venga elaborata.
Siccome 0 ed 1 rappresentano solamente due stati, ci conviene fare dei pacchetti di bit in modo da avere piu' informazioni diverse possibili.

Se invio 1 bit ho 2 diverse informazioni **2^1**
0, 1
Se invio 2 bit ho 4 diverse informazioni **2^2**
00, 01, 10,11
Se invio 3 bit ho 8 possibili informazioni diverse **2^3**
000, 001, 010, 011, 100, 101, 110,111
Se invio 4 bit ho 16 possibili informazioni diverse **2^4**
0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111
...

Quindi, se invio un pacchetto di **n** bit potro' inviare **2^n** informazioni diverse
sono disposizioni con ripetizione di 2 oggetti presi n ad n **$D'_{2;n} = 2^n$**

Dopo varie vicissitudini si scelse di utilizzare per le informazioni pacchetti da **8** bit, quindi un totale di **2^8=256** informazioni diverse.

b) Byte

Si definisce **Byte** un "pacchetto" di 8 bit.
Con 8 bit posso dare al computer 256 informazioni diverse.
Attorno agli anni 70 si scelsero 8 bit per poter implementare sul computer le lettere dell'alfabeto (caratteri estesi) sia maiuscole che minuscole, alcuni simboli matematici, alcuni simboli grafici ed anche dei comandi riservati (byte da 0 a 31; ad esempio il numero decimale 7 fa un suono ed il numero decimale 10 fa avanzare di una riga la macchina da scrivere).
In informatica esiste una legge empirica che afferma:
Ogni anno raddoppia il numero dei componenti in una data area e contemporaneamente il loro prezzo si dimezza.
Questo significa che in informatica avremo continui cambiamenti; inizialmente i pacchetti implementati nei vari computer erano di 5,6,7 bit e dopo gli anni 70 si passo' ai 16 bit ed ai 32 bit, comunque l'unita' di misura fissata resta il Byte e qundo parliamo di un disco fisso di capacita' un terabyte intendiamo un disco su cui possono essere posizionati circa un miliardo di MegaByte (1 MegaByte = 1024 BYte) cioe' piu' di ottomila miliardi di 0 ed 1 (precisamente 1.099.511.627.776 Byte, cioe' 8.796.093.022.208 di interruttori che possono valere 0 od 1).

c) Codice ascii esteso

Attorno agli anni 70 i vari costruttori di computer inventarono dei codici per poter avere i caratteri dell'alfabeto sul computer, ad esempio IBM utilizzo' il codice (EBCDIC Extended Bynari-Codec Decimal Interchanged Code), che essendo utilizzato solamente su macchine IBM non ebbe una difusione universale. Invece il codice che ebbe maggior fortuna fu il codice ASCII (American Standard Code for Information Interchange). Dapprima si ebbe il codice ASCII (7 bit per 128 possibilita') poi il codice ASCII esteso, quello universalmente utilizzato) di 8 bit ed 8 bit = 1 Byte divenne l'unita' di misura dell'informazione.

Cosa c'e' nel codice ASCII esteso; scrivo in numeri decimali.
Le codifiche da **0** a **32** sono "caratteri di controllo" e servono a vari usi, come l'emissione di un suono (7), oppure il ritorno a capo nella macchina da scrivere (13) collegata al computer (ti ricordo che lo schermo e' un'invenzione degli anni 80)
da **33** a **64** vari simboli, quali parentesi tonda aperta (41) e chiusa (42), i numeri decimali da 0 a 9 (da 48 a 57) altri simboli da 58 a 64 tipo = (61), ? (63);
Da **65** a **90** le lettere maiuscole e da **91** a **122** le lettere minuscole

---

La scelta di questi numeri per le lettere e' dovuta al fatto che e' possibile passare dalle lettere maiuscole alle lettere minuscole semplicemente cambiando da **0** ad **1** il terzo bit partendo da sinistra: ad esempio

A = 65_{10} = 01000001_2 a = 97_{10} = 01100001_2
M = 77_{10} = 01001101_2 m = 109_{10}= 01101101_2

da **123** a **127** altri simboli matematici tipo le parentesi graffe (123 e 125);
qui terminavano i caratteri del codice ASCII a 7 bit poi nel codice esteso vennero aggiunti da **128** a **154** caratteri tipografici speciali tipo il dittongo æ (146), È (130), É (138)
da **155** a **175** un misto di simboli matematici e alfabetici, da **176** a **154** ancora simboli grafici e simboli alfabetici speciali, infine c'e' **255** che e' lo spazio vuoto (quello che lasci scrivendo fra una parola e l'altra). Mentre i simboli matematici ed i caratteri tipografici sono universali i simboli grafici che rappresentano i comandi e gli altri simboli grafici possono essere visualizzati diversamente a seconda del software utilizzato per la loro visione.
Per una tabella completa dei caratteri ASCII vi consiglio di consultare Wikipedia
https://it.wikipedia.org/wiki/ASCII
(ricordarsi di costruire una tabella ascii e metterla qua)

Nei vecchi computer era possibile visualizzare i vari simboli corrispondenti ai Byte tenendo premuto il tasto alt e componendo il numero decimale sul tastierino numerico (non sulla tastiera) essendo il tasto BlocNum attivo, al rilascio dei tasti compariva sullo schermo il simbolo relativo.
Ricordo ancora che, se qualcuno voleva smanettare un po' sul computer a livello abbastanza elevato, doveva conoscere il codice a memoria per poter dare le giuste istruzioni alla macchina .
Questo era il problema con i primi computer: senza avere un "traduttore che trasferisse le istruzioni da linguaggio umano a linguaggio macchina era necessario scrivere tutte le istruzioni in bit e, piu' tardi, in linguaggio macchina: ad esempio il comando ritorno carrello (carattere 13 in ASCII 00001101_2 oppure $0D_{16}$ in esadecimale) in linguaggio macchina diventa CR (Carriage Return).
I primi programmatori di solito non resistevano altre i 6 mesi a scrivere direttamente in Byte, poi fu inventato il linguaggio macchina, piu' vicino al linguaggio umano.
Con i nuovi computer e programmi di scrittura e' tutto piu' semplice ed il codice e' piuttosto inutilizzato se non da qualche Haker irriducibile.

Unicode
Il codice Unicode e'un sistema di codifica dei caratteri a 16 bit elaborato nel 1991. Il sistema Unicode permette di rappresentare tutti i caratteri attraverso un codice a 16 bit, indipendentemente da qualsiasi sistema operativo o linguaggio di programmazione. Esso raggruppa infatti la quasi totalita' degli alfabeti esistenti (arabo, armeno, cirillico, greco....)

8. Sistema ottale

Un certo interesse in informatica ha il sistema di numerazione a base 8.
In pratica non e' molto usato; noi qui lo ricordiamo solo per una sua peculiare caratteristica.
Le cifre binarie corrispondenti alle cifre ottali sono date da soli 3 bit; infatti abbiamo, per le cifre possibili:

Cifre ottali	Cifre binarie
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

La caratteristica interessante e' quindi che ogni un numero a base 8 puo' essere trasformato immediatamente in binario sostituendo ad ogni sua cifra ottale le tre cifre binarie corrispondenti.
Per noi esseri umani e' piu' facile trasformare un numero dal sistema decimale nel sistema ottale e quindi diventa automatico trasformarlo poi in binario.

Esempio: trasformare il numero $\mathbf{6301_8}$ nel suo equivalente binario.
Metto al posto di ogni cifra la tripletta di bit equivalente:

6	3	0	1
110	011	000	001

Quindi:
$\mathbf{6301_8 = 110011000001_2}$

Esercizio: il bilancio di un ministero degli Stati Uniti e' stato nel 1946 di 1.426.895.325 $.
Inserire tale cifra in un calcolatore elettronico significa trasformare quella cifra in binario.
Prima la trasformo in ottale dividendo per 8 e considerando i resti
1426895325 : 8 da' 178361915 con resto di 5
178.361.915 : 8 da' 22295239 con resto di 3
22295239 : 8 da' 2786904 con resto di 7
2786904 : 8 da' 348363 con resto 0
348363 :8 da' 43535 con resto di 3
43535 : 8 da' 5441 con resto di 7
5441 : 8 da' 680 con resto di 1
680 : 8 da' 85 con resto 0
85 ; 8 da' 10 con resto di 5
10 : 8 da' 1 con resto di 2
1 : 8 da' 0 con resto di 1
Quindi, riscrivendo i resti cominciando dall'ultimo fino al primo
$\mathbf{1426895325_{10} = 12501730735_8 = 001010101001111011000111001101_2 =}$
$\mathbf{1010101001111011000111001101_2}$.

Spero vi siate convinti che, per un essere umano, e' piu' semplice trasformare cosi' piuttosto che trasformare subito in base 2.

9. Sistema esadecimale

a) Sistema di numerazione a base 16

Importantissimo in informatica, il sistema a base 16 (esadecimale) ha bisogno di 16 cifre diverse per poter rappresentare i numeri da 1 a 15; al sedici andremo "a capo" per formare la "sedicina".
Siccome le cifre numeriche sono solamente 9, per le altre cifre chiederemo aiuto alle lettere maiuscole dell'alfabeto latino ed avremo quindi:

Cifre decimali	Cifre esadecimali
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	A
11	B
12	C
13	D
14	E
15	F
16	10
....	....
25	19
26	1A
....	....
31	1F
32	20
....	....
255	FF
256	100
....	....

In informatica la trasformazione da numero decimale qualunque a numero esadecimale e' piuttosto poco importante come anche il trasformare un numero esadecimale qualunque in decimale; invece la cosa diventa importantissima se il numero decimale e' inferiore od uguale a 255 (se il numero esadecimale e' inferiore od uguale ad FF).

b) Relazioni fra sistema esadecimale e binario

Vale la relazione importantissima:
Un numero esadecimale si puo' trasformare in binario semplicemente trasformando in binario ciascuna delle sue cifre.

Ad esempio il numero esadecimale **F A** corrispondera' al binario **1111 1010**

Da notare che un esadecimale a due cifre e' composto da 8 bit ed 8 bit = 1 Byte e' il pacchetto di base scelto per trasmettere le informazioni (codice ASCII)
Quindi, la memoria del nostro computer si potra' rappresentare mediante blocchi esadecimali a due cifre, molto piu' leggibili che blocchi di 8 bit.

Per poter fare qualcosa di buono e' necessario conoscere a memoria l'equivalenza fra cifra esadecimale e Byte per le cifre da 0 a F(15)

Numero decimale	Cifra esadecimale	Byte corrispondente
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Detto cio' sara' immediato associare ad ogni carattere ASCII una coppia di numeri esadecimali da 00 (per il Byte 00000000) ad FF (per il Byte 11111111).

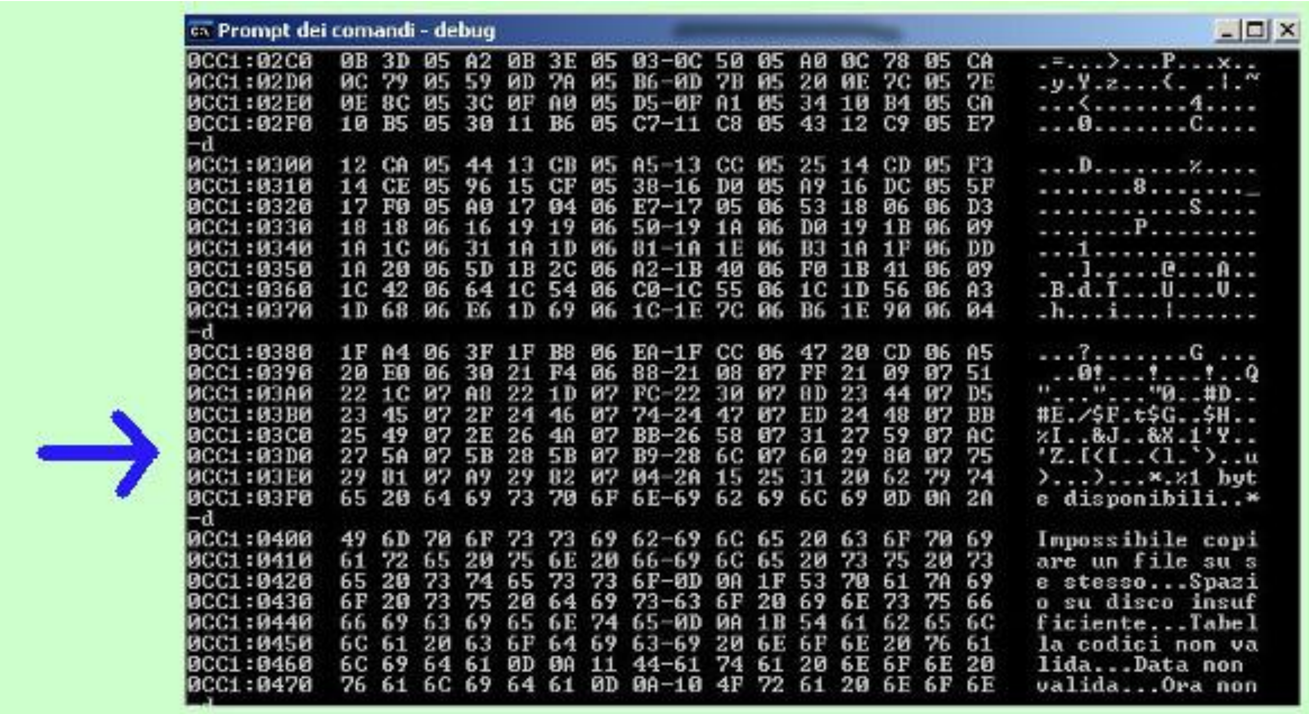
c) Un esempio: il comando "debug"

Come esempio di quanto detto proviamo il comando debug.

Premetto che tale comando e' funzionante solamente su sistemi derivanti dal DOs e quindi su Windows; se volete attivarlo in linux dovete avere il programma di emulazione Wine installato e funzionante.
Ricordo inoltre che i caratteri ASCII hanno grafica e utilizzi diversi a seconda del sistema operativo utilizzato e quindi su macchine diverse potreste vedere in modo diverso rappresentati i caratteri di controllo
Raccomando inoltre di seguire esattamente le istruzioni e non provare a "smanettare" a caso: potreste DANNEGGIARE il vostro sistema.

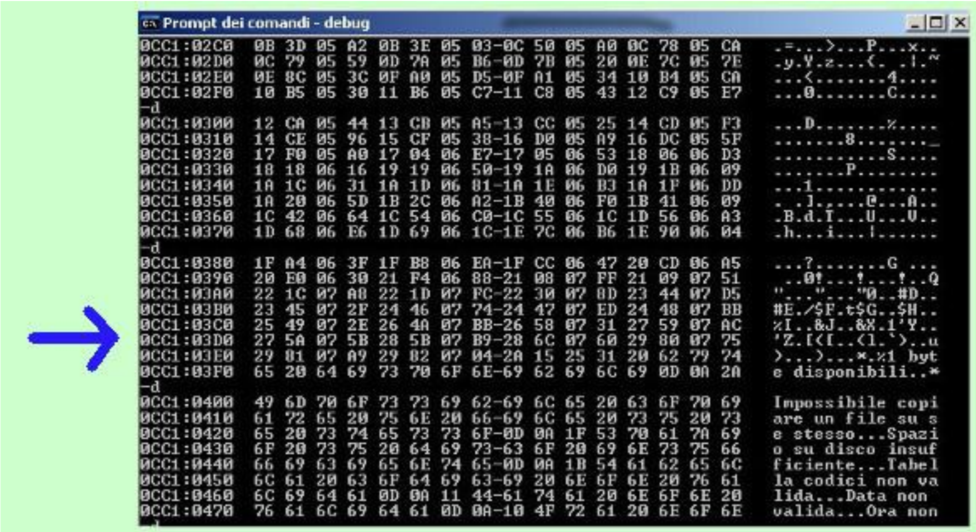
In Windows aprite "tutti i programmi", scegliete "accessori" e scegliete l'opzione "terminale".
Vi si aprira' una finestra in bianco e nero con il prompt dei comandi:
digitate **debug** e premete invio;
otterrete una breve lineetta lampeggiante
digitate **d** (comando dump = salta) e premete invio; se ottenete tutti 0 digitate ancora **d** ed ogni volta premete invio.
Per uscire scrivete semplicemente **q** (comando quit = esci) e premete invio.
Poi, per chiudere il terminale scrivete **exit** e premete ancora il tasto invio

Digitando **d** piu' volte otterrete una schermata simile a questa



Questo e' il vostro computer: nelle prime colonne ci sono gli indirizzi a 32 bit dei dati della macchina.
Dovete pensare al computer come ad un immenso nastro dove in ogni posizione (indirizzo) c'e' un dato (Byte) ed ogni posizione e' data da un numero binario a 16 bit.
Intuitivamente il nastro scorre ed i dati possono essere letti ed utilizzati da un lettore.
Come esempio consideriamo la sesta riga del secondo blocco intero (18-sima riga dell'immagine dove ho posizionato la freccia).
L'indirizzo cui sono scritti questi dati e', in esadecimale, da 0CC103D0 a 0CC103DF (su ogni riga e' scritto solamente l'indirizzo di inizio); quindi, se volessimo scrivere l'indirizzo in

binario questi sono i dati:
dall'indirizzo:
0000 1100 1100 0001 0000 0011 1101 0000
all'indirizzo:
0000 1100 1100 0001 0000 0011 1101 1111
In questi 16 indirizzi in ordine ci sono 16 dati in esadecimale:
27 5A 07 5B 28 5B 07 B9 28 6C 07 60 29 80 07
I 16 gruppi di 2 numeri in esadecimale corrispondono ai caratteri ascii della parte destra dello schermo:



Cioe' sarebbe a dire che all'indirizzo a sinistra corrisponde l'insieme di dati al centro, insieme di dati che a destra e' in codice ASCII.
Ricordo che nell'indirizzo sono sottointesi gli indirizzi compresi fra quello indicato e quello della riga sotto (16 indirizzi piu' avanti)

indirizzo esadecimale	indirizzo binario	dato in esadecimale	dato in Byte
0CC103D0	0000 1100 1100 0001 0000 0011 1101 0000	27	00100110
0CC103D1	0000 1100 1100 0001 0000 0011 1101 0001	5A	01011010
0CC103D2	0000 1100 1100 0001 0000 0011 1101 0010	07	00000111
0CC103D3	0000 1100 1100 0001 0000 0011 1101 0011	5B	01011011
0CC103D4	0000 1100 1100 0001 0000 0011 1101 0100	28	00101000
0CC103D5	0000 1100 1100 0001 0000 0011 1101 0101	5B	01011011
0CC103D6	0000 1100 1100 0001 0000 0011 1101 0110	07	00000111
0CC103D7	0000 1100 1100 0001 0000 0011 1101 0111	B9	10111001
0CC103D8	0000 1100 1100 0001 0000 0011 1101 1000	28	00101000
0CC103D9	0000 1100 1100 0001 0000 0011 1101 1001	6C	011001100
0CC103DA	0000 1100 1100 0001 0000 0011 1101 1010	07	00000111
0CC103DB	0000 1100 1100 0001 0000 0011 1101 1011	60	01100000
0CC103DC	0000 1100 1100 0001 0000 0011 1101 1100	29	00101001
0CC103DD	0000 1100 1100 0001 0000 0011 1101 1101	80	10000000
0CC103DE	0000 1100 1100 0001 0000 0011 1101 1110	07	00000111
0CC103DF	0000 1100 1100 0001 0000 0011 1101 1111	75	01110101
0CC103E0	primo indirizzo della riga sotto		

In pratica il computer scrive cosi', intercalando i dati con bit di controllo (chiamiamoli car.c.); io ti scrivo i primi 5 byte, lascio qualche spazio per far seguire meglio, e distinguo nel colore i 32 bit di indirizzo (in verde) dagli 8 bit di dati (in rosso) ma nel computer non ci sono spazi:
(car.c.) 0000 1100 1100 0001 0000 0011 1101 0000 00100110 (car.c.)0000 1100 1100 0001 0000 0011 1101 0001 01011010 (car.c.) 0000 1100 1100 0001 0000 0011 1101 0010 00000111 (car.c.) 0000 1100 1100 0001 0000 0011 1101 0011 01011011 (car.c.) 0000 1100 1100 0001 0000 0011 1101 0100 00101000 (car.c.) .....

Ad esempio, il primo dato 27_{16} corrisponde al carattere decimale 39 (16x2+9) cui corrisponde il carattere ascii **'** (apostrofo);
il secondo dato $5A_{16}$ corrisponde al carattere decimale 90 (16x5+10) cui corrisponde il carattere ascii Z;
il terzo dato 07_{16} e' un comando e precisamente quello che da' il suono del campanello e a destra viene evidenziato con un puntino; e siccome sulla riga ce ne sono 4 piu' uno sotto penso che abbiamo "beccato" dove Windows tiene il suono del campanello per segnalare gli errori (din,din din din din);
tutti i comandi e i caratteri diversi dai tipografici vengono evidenziati con un puntino, ad eccezione del 20_{16} che ci da' lo spazio tra una parola e l'altra.

Potete divertirvi a guardare, nel blocco successivo, alcuni dei messaggi di errore piu' comuni che restituisce Windows.

Quando (sistemi operativi Spectrum, Commodore, Amiga, DOS,...) i giochi su computer erano carissimi e rarissimi spesso qualcuno passava le nottate sul debug per poter individuare la password che permetteva di entrare nel gioco oppure di superare lo sbarramento di un livello (errori di gioventu'.....)

C. CENNI SULL'ALGEBRA BINARIA DI BOOLE

1. Un cenno storico

George Boole fu uno dei grandi matematici dell'ottocento, studioso teorico dell'algebra astratta applicata alla logica; la costruzione che a noi interessa si riferisce ad una struttura particolare: qual'e' la possibile struttura algebrica che operi su un numero di due oggetti diversi seguendo le regole della logica?
Caso particolare delle algebre di Boole, le algebre binarie impostano il ragionamento su due oggetti diversi (pari o dispari, zero od uno, vero o falso).
A quei tempi, con i matematici volti soprattutto alla strutturazione dell'analisi matematica, le algebre di Boole (o algebre logiche) considerate piu' che altro un gioco matematico e furono presto dimenticate, sinche', nel novecento, si arrivo' a considerare l'informatica come l'insieme di operazioni logiche su due oggetti specifici in concomitanza con il calcolo binario: l'informatica aveva trovato le sue basi per poter crescere e svilupparsi non solo come pratica ma anche come teoria

2. Cos'e' una struttura algebrica

Vedi anche i concetti espressi in algebra astratta .

Vediamo prima, con qualche esempio, di chiarire il concetto di struttura algebrica:

Primo esempio:
Considero due insiemi: l'insieme dei numeri pari e l'insieme dei numeri dispari
Se applico l'operazione di somma normale fra numeri naturali avro':
pari + pari = pari

pari + dispari = dispari
dispari + pari = dispari
dispari + dispari = pari
Se applico la normale operazione di prodotto fra numeri naturali avro':
pari x pari = pari
pari x dispari = pari
dispari x pari = pari
dispari x dispari = dispari

Secondo esempio:
considero l'insieme dei resti modulo 2 (relazione di congruenza modulo 2)
Se applico l'operazione di somma avro':
$0 \oplus 0 = 0$
$0 \oplus 1 = 1$
$1 \oplus 0 = 1$
$1 \oplus 1 = 0$
se applico l' operazione di prodotto avro':
$0 \otimes 0 = 0$
$0 \otimes 1 = 0$
$1 \otimes 0 = 0$
$1 \otimes 1 = 1$

Mi sembra chiaro che esiste qualcosa che lega gli esempi considerati: i due esempi hanno "strutture" simili, cioe' le operazioni si comportano in modo similare anche se gli oggetti su cui operano sono diversi.

Vediamo altri due esempi.

Esempio 3
Considero i numeri positivi e negativi, con l'operazione di prodotto; avro':
positivo $\otimes$ positivo = positivo
positivo $\otimes$ negativo = negativo
negativo $\otimes$ positivo = negativo
negativo $\otimes$ negativo = positivo
Esempio 4
Suddividiamo gli esseri umani in amici e nemici. Considero l'operazione "del":
l'amico dell'amico e' un amico
l'amico del nemico e' un nemico
il nemico dell'amico e' un nemico
il nemico del nemico e' un amico

Anche qui qualcosa lega gli esempi considerati, inoltre il nostro concetto di "operazione" (chiamiamola legge di composizione interna) si e' fatto piu' ampio.

Si definisce **struttura algebrica** un insieme non vuoto **A** su cui siano definite una o piu' leggi di composizione interna.
Cioe': struttura algebrica = insieme con operazione (i), poi il tipo di struttura dipendera' dalle proprieta' delle operazioni; indicheremo una struttura algebrica nei seguenti modi:
$(A; \oplus, \otimes)$ Struttura con due leggi di composizione
$(A; \textcircled{T})$ Struttura con una legge di composizione

Concludendo: considerando un insieme di enti, e' importante vedere quali operazioni sono possibili e le varie proprieta' che hanno queste operazioni in tali insiemi; in tal modo potremo individuare delle strutture che ci permetteranno di classificare gli insiemi a seconda delle proprieta' comuni che hanno.
Nella prossime pagine, dopo un breve approfondimento sulle operazioni, vedremo una struttura con un insieme composto dai soli elementi 0 ed 1 (algebra binaria di Boole)

3. Operazioni interne

Prima di procedere approfondiamo un poco il concetto di operazione.
Se vuoi far contento il tuo Prof. puoi indicarla, in linguaggio erudito, come legge di composizione
Vedi anche la pagina in aritmetica.

Intuitivamente un'operazione e' qualunque cosa che applicata ad uno o piu' oggetti li trasforma in qualcosa.

Sono applicazioni:
nell'insieme dei numeri interi l'opposto di un numero   opposto(+3)=-3
nell'insieme dei numeri naturali la somma di due numeri   somma(3,4)= 3+4 = 7
nei libri di fiabe il bacio di una principessa   bacio di principessa(rospo)=principe
l'applicazione identica trasforma un qualunque oggetto in se' stesso
identita'(a) = a.

Come prima cosa distinguiamo le applicazioni a seconda del numero di oggetti su cui operano; avremo:
Operazione unaria se applicata su un oggetto restituisce un oggetto
Operazione binaria se applicata su due oggetti restituisce un oggetto
Operazione ternaria se applicata su tre oggetti restituisce un oggetto
....
Inoltre, potremo distinguere le applicazioni che trasformano oggetti di un insieme in un elemento dello stesso insieme (**interne**) e le applicazioni che trasformano oggetti di un insieme in un elemento di un altro insieme (**non interne**)

Esempi
Nell'insieme dei numeri naturali la somma e' un'operazione binaria interna, mentre la differenza no (basta che esista almeno un elemento che non appartenga e l'operazione e' detta non interna):
3 + 2 = 5 interna
5 - 3 = non si puo' fare in **N** quindi non interna.
Nell'insieme dei numeri razionali privati dello zero **{Q-0}** l'operazione di inverso e' un'operazione unaria interna, mentre non lo e' nell'insieme dei numeri interi:
in **{Q-0} inv(2/3)= 3/2**
in **N inv(3)** non esiste.
Esercizio per rilassarsi: nell'insieme di rospi il bacio di una principessa e' un'operazione interna?
Risposta: solamente se il principe e' molto, molto, molto brutto

A noi, interessano, per l'algebra di Boole, 3 operazioni, tutte e tre interne.
La prima operazione unaria: **passaggio al complementare** che indicheremo con l'apostrofo **'**

Due operazioni binarie:
- una l'indicheremo come **somma** (attenzione! e' diversa dalla somma che conosciamo) e, nelle seguenti pagine di teoria useremo il simbolo $\oplus$, poi nella pratica useremo semplicemente il **+**

- l'altra l'indicheremo come **prodotto** (attenzione! e' diversa dal prodotto che conosciamo) e, nelle seguenti pagine di teoria, useremo il simbolo $\otimes$; poi nella pratica useremo semplicemente il **·, e, dove non ci saranno possibilita' di errori, lo sottointenderemo.**

4. Algebra binaria di Boole

Consideriamo un insieme **B** in cui siano definite un'operazione unaria indicata con ' e due operazioni binarie indicate con $\otimes$, $\oplus$ che agisca su due oggetti distinti 0 ed 1.

$$\{ B, ', \otimes, \oplus; 0, 1 \}$$

Chiamiamo a, b, c, d,.... degli elementi dell'insieme B
Quindi a puo' essere 0 od 1, b puo' essere 0 od 1, c puo' essere 0 od 1, d puo' essere 0 od 1, eccetera
Questo insieme e' detto algebra di Boole se valgono le seguenti leggi

- *Legge commutativa*
  $a \oplus b = b \oplus a$
  $a \otimes b = b \otimes a$
- *Legge distributiva*
  $a \oplus (b \otimes c) = (a \oplus b) \otimes (a \oplus c)$ notare che questa e' molto diversa dalle leggi per la somma normale in N
  $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$
- *Leggi dell'identita'*
  $a \oplus 0 = a$ cioe' **0** e' l'elemento neutro per $\oplus$
  $a \otimes 1 = a$ cioe' **1** e' l'elemento neutro per $\otimes$
- *Leggi del complemento* significano semplicemente che se **a** vale **0** allora **a'** vale **1** e se **a** vale **1** allora **a'** vale **0**
  $a \oplus a' = 1$
  $a \otimes a' = 0$

5. Ordine delle operazioni

Abbiamo le possibili operazioni:

1. *Passaggio al complementare*
   Corrisponde alla negazione fra elementi logici (not)

   0'=1
   1'=0

a	a'
1	0
0	1

2. *Somma*
   Non corrisponde alla nomale somma fra numeri naturali, ma e' somma fra elementi logici (vel; or logico)

   0 + 0 = 0
   0 + 1 = 1
   1 + 0 = 1
   1 + 1 = 1

+	0	1
0	0	1
1	1	1

3. *Prodotto:*
Non corrisponde al normale prodotto fra numeri naturali, ma e' un prodotto fra elementi logici (and logico)

$0 \cdot 0 = 0$
$0 \cdot 1 = 0$
$1 \cdot 0 = 0$
$1 \cdot 1 = 1$

·	0	1
0	0	0
1	0	1

Sorge il problema; quando abbiamo un'espressione di quale operazione va fatta prima e quale dopo se il testo non e' chiaro.
Seguiremo questo ordine:
prima va fatto il passaggio al complementare, poi il prodotto ed infine la somma e, per variare l'ordine, useremo le parentesi come facciamo nelle normali espressioni numeriche.
Vediamo qualche esempio; io faccio tutti i passaggi, ma si puo' abbreviare (nei passaggi, se fermi il mouse su ogni risultato, potrai vedere quale proprieta' ho applicato per trovarlo).

Calcolare, per quanto possibile l'espressione:
$a + a \cdot b + b + a \cdot b + b' =$
$= a + (a \cdot b + a \cdot b) + (b + b') =$
$= a + a \cdot b + 1 =$
$= a \cdot (1+b) + 1 =$
$= a \cdot 1 + 1 =$
$= a + 1 =$
$= 1$

Calcolare, per quanto possibile, l'espressione
$a + a' \cdot (a+b) + b \cdot (a'+b) + a \cdot b' + b' \cdot (a+b') =$
$= a + a' \cdot a + a' \cdot b + b \cdot a' + b^2 + a \cdot b' + b' \cdot a + b'^2 =$
$= a + 1 + (a' \cdot b + b \cdot a') + b + (a \cdot b' + b' \cdot a) + b' =$
$= a + 1 + (a' \cdot b + a' \cdot b) + b + (a \cdot b' + a \cdot b') + b' =$
$= a + 1 + a' \cdot b + b + a \cdot b' + b' =$
$= a + 1 + (a' \cdot b + b) + (a \cdot b' + b') =$
$= a + 1 + b \cdot (a'+1) + b' \cdot (a' + 1) =$
$= a + 1 + b \cdot 1 + b' \cdot 1 =$
$= a + 1 + b + b' =$
$= a + 1 + (b + b') =$
$= a + 1 + 1 =$
$= a + (1+1) =$
$= a + 1 =$
$= 1$

Siccome 1 sommato a qualunque espressione da' sempre 1, potevamo calcolare il risultato sia di questa che della precedente, gia' dal passaggio in cui e' comparso l'1; pero' ho continuato per farti vedere i vari passaggi.

6. Principio di dualita'

In un'algebra di Boole la duale di una qualsiasi asserzione e' l'asserzione che si ottiene scambiando fra loro il prodotto con la somma e lo 0 con l'1.
Esempio, l'uguaglianza:
$(a+1)\cdot(b+0) = b$
si puo' trasformare per dualita' nell'uguaglianza:
$(a\cdot0)+(b\cdot1) = b$
In effetti se guardi le leggi (assiomi) dell'algebra binaria di Boole, vedi che le proprieta' elencate sono duali fra loro.
Vale il teorema (principio di dualita')
Il duale di un teorema nell'algebra di Boole e' ancora un teorema.

Notiamo la stretta corrispondenza esistente fra informatica, teoria degli insiemi e logica: in teoria degli insiemi hai le leggi di De Morgan, qui hai il principio di dualita' che non sono altro che lo stesso tipo di struttura su aspetti diversi della stessa realta'.

---

7. Alcuni teoremi

Vediamo assieme alcuni teoremi che saranno utili per i calcoli.
Per ogni teorema indicheremo anche il duale.
Dimostrato il teorema non ci sarebbe bisogno di dimostrare il duale, perche', per il principio di dualita' fatto nella pagina precedente, se e' vero il teorema e' vero anche il suo duale; pero' il farlo puo' essere un utile esercizio.

Purtroppo in algebra astratta non possiamo avere l'intuizione che abbiamo nella matematica piu' convenzionale, quale analisi o geometria del piano, inoltre le operazioni che consideriamo le chiamiamo somma e prodotto solamente per convenienza, ma possono essere molto diverse dalla somma e dal prodotto che conosciamo; ogni costruzione di algebra astratta obbedisce solamente ai postulati iniziali e, variando un postulato, possiamo ottenere risultati molto diversi ed inaspettati.
Tutto questo per dire che nelle dimostrazioni, non solo vanno considerati **tutti** i passaggi, senza poterne saltare nessuno, ma ogni passaggio deve essere **sempre giustificato** evidenziando la regola che lo permette: tutto cio' rende piuttosto pesante sia la dimostrazione dei teoremi che lo sviluppo degli esercizi (fra dimostrazione di teoremi e sviluppo di esercizi non vi e' molta differenza).
Possiamo comunque considerare la dimostrazione di teoremi come degli esercizi e comportarci di conseguenza.
Purtroppo tutto cio' rende l'algebra astratta un qualcosa di molto pesante, almeno dal puto di vista dei passaggi.
Purtroppo non c'e' niente da fare: e' il prezzo che si deve pagare per essere sicuri di avere un ragionamento rigoroso ed esatto.
Comunque, per non appesantire troppo, io mi limito qui a dimostrare i primi 3 teoremi per gli altri forse in futuro.....

- ***Idempotenza***
  $a + a = a$ duale $a \cdot a = a^2 = a$
  Dimostrazione:
  Purtroppo in algebra astratta non possiamo avere l'intuizione che abbiamo nella matematica piu' convenzionale, quale analisi o geometria del piano, inoltre le operazioni che consideriamo le

chiamiamo somma e prodotto solamente per convenienza, ma possono essere molto diverse dalla somma e dal prodotto che conosciamo; ogni costruzione di algebra astratta obbedisce solamente ai postulati iniziali e, variando un postulato, possiamo ottenere risultati molto diversi ed inaspettati.
Tutto questo per dire che nelle dimostrazioni, non solo vanno considerati **tutti** i passaggi, senza poterne saltare nessuno, ma ogni passaggio deve essere sempre giustificato con il ricorso alla regola che lo permette: tutto cio' rende piuttosto pesante sia la dimostrazione dei teoremi che lo sviluppo degli esercizi (fra dimostrazione di teoremi e sviluppo di esercizi non vi e' molta differenza).
Possiamo comunque considerare questa pagina e le altre con la dimostrazione di teoremi come degli esercizi e comportarci di conseguenza.
Purtroppo tutto cio' rende l'algebra astratta un qualcosa di molto pesante, almeno dal puto di vista dei passaggi: vedi ad esempio, piu' avanti, la dimostrazione della legge associativa.
Purtroppo non c'e' niente da fare: e' il prezzo che si deve pagare per essere sicuri di avere una matematica rigorosa ed esatta .

Devo dimostrare la legge dell'idempotenza.
Faccio riferimento alle leggi di definizione dell'algebra di Boole; a destra ti indico la legge applicata per ottenere il risultato:
$a + a = a$
Parto da **a**, so che:
$a = a + 0$ (prima legge dell'identita')
$a + 0 = a + (a \cdot a')$ (seconda legge del complemento)
$a + (a \cdot a') = (a + a) \cdot (a + a')$ (prima legge distributiva)
$(a + a) \cdot (a + a') = (a + a) \cdot 1$ (prima legge del complemento)
$(a + a) \cdot 1 = a + a$ (seconda legge dell'identita')
Quindi, per la proprieta' transitiva delle uguaglianze, leggendo l'ultimo ed il primo termine delle uguaglianze otteniamo:
$a + a = a$
Come volevamo.

Dimostriamo anche la formula complementare: nota che la dimostrazione e' la stessa cambiando il prodotto in somma, cambiando lo 0 in 1 e considerando la stessa legge ma con numero diverso: seconda al posto della prima e prima al posto della seconda.
tenendo presente cio', puoi fare tu la dimostrazione complementare e controllare poi i passaggi cosi' ti serve di esercizio anche per ripassare le regole.

$a \cdot a = a$
Parto da **a**, so che:
$a = a \cdot 1$ (seconda legge dell'identita')
$a \cdot 1 = a \cdot (a + a')$ (prima legge del complemento)
$a \cdot (a + a') = (a \cdot a) + (a \cdot a')$ (seconda legge distributiva)
$(a \cdot a) + (a \cdot a') = (a \cdot a) + 0$ (seconda legge del complemento)
$(a \cdot a) + 0 = a \cdot a$ (prima legge dell'identita')
Quindi, per la proprieta' transitiva delle uguaglianze, leggendo l'ultimo ed il primo termine delle uguaglianze otteniamo:
$a \cdot a = a$
come volevamo

- *legge dei confini*
  $a + 1 = 1$ duale $a \cdot 0 = 0$
  Dimostrazione:
  Devo dimostrare la legge dei confini.
  Faccio riferimento alle leggi di definizione dell'algebra di Boole; a destra ti indico la legge applicata per ottenere il risultato:
  $a + 1 = 1$

Parto da $\mathbf{a + 1}$, so che:
$\mathbf{a + 1 = (a + 1)\cdot 1}$ (seconda legge dell'identita')
$\mathbf{(a + 1)\cdot 1 = (a + 1)\cdot(a + a')}$ (prima legge del complemento)
$\mathbf{(a + 1)\cdot(a + a') = a + (1 \cdot a')}$ (prima legge distributiva letta a rovescio)
$\mathbf{a + (1 \cdot a') = a + a'}$ (seconda legge dell'identita')
$\mathbf{a + a' = 1}$ (prima legge del complemento)
Quindi, per la proprieta' transitiva delle uguaglianze, leggendo il primo e l'ultimo termine delle uguaglianze otteniamo:
$\mathbf{a + 1 = 1}$
Come volevamo.

Dimostriamo anche la formula complementare: nota che la dimostrazione e' la stessa cambiando il prodotto in somma, cambiando lo 0 in 1 e considerando la stessa legge ma con numero diverso: seconda al posto della prima e prima al posto della seconda.
tenendo presente cio', puoi fare tu la dimostrazione complementare e controllare poi i passaggi cosi' ti serve di esercizio anche per ripassare le regole.

$\mathbf{a \cdot 0 = 0}$
Parto da $\mathbf{a \cdot 0}$, so che
$\mathbf{a \cdot 0 = (a\cdot 0) + 0}$ (prima legge dell'identita')
$\mathbf{(a\cdot 0) + 0 = (a\cdot 0) + (a\cdot a')}$ (seconda legge del complemento)
$\mathbf{(a\cdot 0) + (a\cdot a') = a\cdot(0 + a')}$ (seconda legge distributiva letta a rovescio)
$\mathbf{a\cdot(0 + a') = a\cdot(a' + 0)}$ (proprieta' commutativa della somma)
$\mathbf{a\cdot(a' + 0) = a\cdot a'}$ (prima legge dell'identita')
$\mathbf{a\cdot a' = 0}$ (seconda legge del complemento)
Quindi, per la proprieta' transitiva delle uguaglianze, leggendo il primo e l'ultimo termine delle uguaglianze otteniamo:
$\mathbf{a \cdot 0 = 0}$
Come volevamo.

- ***assorbimento***
  $\mathbf{a + (a\cdot b) = a}$ duale $\mathbf{a\cdot(a + b) = a}$
  Dimostrazione.
  Devo dimostrare la legge dell'assorbimento.
  Faccio riferimento alle leggi di definizione dell'algebra di Boole; a destra ti indico la legge applicata per ottenere il risultato:
  $\mathbf{a + (a\cdot b) = a}$
  Parto da $\mathbf{a + (a\cdot b)}$, so che:
  $\mathbf{a + (a\cdot b) = (a\cdot 1) + (a\cdot b)}$ (seconda legge dell'identita')
  $\mathbf{(a\cdot 1) + (a\cdot b) = a \cdot(1 + b)}$ (seconda legge distributiva letta a rovescio)
  $\mathbf{a \cdot(1+b) = a \cdot(b+1)}$ (prima proprieta' commutativa)
  $\mathbf{a \cdot(b+1) = a \cdot 1}$ (legge dei confini dimostrata prima)
  $\mathbf{a \cdot 1 = a}$ (seonda legge dell'identita')
  Quindi, per la proprieta' transitiva delle uguaglianze, leggendo il primo e l'ultimo termine delle uguaglianze otteniamo:
  $\mathbf{a + (a\cdot b) = a}$
  Come volevamo.

Dimostriamo anche la formula complementare: nota che la dimostrazione e' la stessa cambiando il prodotto in somma, cambiando lo 0 in 1 e considerando la stessa legge ma con numero diverso: seconda al posto della prima e prima al posto della seconda
tenendo presente cio', puoi fare tu la dimostrazione complementare e controllare poi i passaggi cosi' ti serve di esercizio anche per ripassare le regole

$a \cdot (a + b) = a$
parto da $a \cdot (a + b)$, so che:
$a \cdot (a + b) = (a + 0) \cdot (a + b)$ (prima legge dell'identita')
$(a + 0) \cdot (a + b) = a + (0 \cdot b)$ (prima legge distributiva letta a rovescio)
$a + (0 \cdot b) = a + (b \cdot 0)$ (proprieta' commutativa del prodotto)
$a + (b \cdot 0) = a + 0$ (legge dei confini dimostrata prima)
$a + 0 = a$ (prima legge dell'identita')
Quindi, per la proprieta' transitiva delle uguaglianze, leggendo il primo e l'ultimo termine delle uguaglianze otteniamo:
$a \cdot (a + b) = a$
come volevamo.

- ***associativita'***
  $(a + b) + c = a + (b + c)$

- ***unicita' del complemento***
  se $a + x = 1$ ed $a \cdot x = 0$ allora $x = a'$

- ***legge del doppio complemento***
  $((a)')' = a$

8. Ordinamento parziale

E' fondamentale in informatica considerare l'ordine in cui vengono date le informazioni: ho gia' detto che, semplificando, e' come un lungo nastro di dati che scorre davanti ad un lettore, di conseguenza i dati dovranno essere ordinati per poter essere utilizzati

---

Ripartiamo dal concetto di relazione d'ordine e vediamo il concetto di **ordine parziale**.
Diremo che una relazione **R** su un insieme **A** e' una relazione di **ordine** se e':
- riflessiva
- antisimmetrica
- transitiva

E qui sarebbe meglio ripassare tutto il capitolo sulle relazioni

---

Diremo che una relazione e' di **ordine totale** su un insieme **B,** se dati comunque due elementi **a** e **b** appartenenti a **B** vale sempre:
o aRb oppure bRa
Cioe' tutti gli elementi sono confrontabili fra loro secondo la relazione.

Esempio: l'insieme dei numeri naturali **N** con la relazione **<** e' un insieme totalmente ordinato perche' per due qualunque numeri naturali **a** e **b** posso sempre dire se vale **a<b** oppure **b<a**

Se pero' esistono elementi di **B** fra loro non confrontabili secondo la relazione, allora la relazione si dice di **ordine parziale** e l'insieme **B** si dice **parzialmente ordinato .**

L'insieme degli esseri umani con la relazione **"e' antenato o identico di"** e' solo di ordine parziale perche' due che siano fratelli non appartengono alla relazione.

Ripassiamo le relazioni nei particolari:

I. ***riflessiva:***
$a \leq a \quad \forall a \in B$
Si legge: **a** e' minore od uguale ad **a** per ogni elemento **a** appartenente a **B**
cioe', in **B** ogni elemento e' minore od uguale a se' stesso (oppure, che e' la stessa cosa, non e' superiore a se' stesso)

II. ***antisimmetrica:***
se $a \leq b$ e $b \leq a \Rightarrow a=b \quad \forall a,b \in B$
Si legge: se **a** e' minore od uguale a **b** e **b** e' minore od uguale ad **a**, allora **a** e' uguale a **b** per ogni coppia di elementi **a,b** appartenenti a **B**
cioe' se **a** non e' maggiore di **b** e **b** non e' maggiore di **a** allora **a** e **b** sono uguali

III. ***transitiva:***
$a \leq b,\ b \leq c \Rightarrow a \leq c \quad \forall a,b,c \in B$
Si legge: se **a** e' minore od uguale a **b** e **b** e' minore od uguale a **c** allora **a** e' minore od uguale a **c** per ogni terna di elementi **a,b,c** appartenenti a **B**

Le algebre di Boole sono parzialmente ordinate a causa del teorema:
Sia B un' algebra di Boole, allora B e' un insieme parzialmente ordinato con $a \leq b$ se e solo se $a+b=b$

Dimostriamolo "alla buona" con una dimostrazione non rigorosa, ma efficace, valida solo per l'algebra binaria di Boole (abbiamo solo gli elementi **0** ed **1**).
La condizione e' necessaria e sufficiente (corrisponde a **se e solo se**) prima consideriamo la dimostrazione diretta.
Ipotesi: supponiamo che nell'algebra binaria valga $\mathbf{a + b = b}$
Tesi: devo dimostrare che vale $\mathbf{a \leq b}$
L'ipotesi significa: $\mathbf{0 + 0 = 0}$; $\mathbf{0 + 1 = 1}$; $\mathbf{1 + 1 = 1}$
cioe' la somma equivale al secondo addendo se sono contemporaneamente nulli entrambe gli addendi oppure se il secondo addendo vale 1;
di conseguenza segue per ogni elemento **a** (che puo' essere **0** od **1**
se **a=0 e se b=0** $\quad \mathbf{0 \leq 0}$ cioe' $\mathbf{a \leq b}$
se **a=0 e se b=1** $\quad \mathbf{0 \leq 1}$ cioe' $\mathbf{a \leq b}$
se **a=1 e se b=1** $\quad \mathbf{1 \leq 1}$ cioe' $\mathbf{a \leq b}$
quindi:
$\mathbf{a \leq b}$

Dimostrazione inversa:
Ipotesi: supponiamo che in **B** valga $\mathbf{a \leq b}$
Tesi: devo dimostrare che vale $\mathbf{a + b = b}$
Gli unici elementi dell'insieme **B** sono **0** ed **1**
Vale: $\mathbf{0 \leq 0}$, $\mathbf{0 \leq 1}$, $\mathbf{1 \leq 1}$,
quindi si ha, considerando i casi possibili:
se **a=0 e se b=0 0+0=0=a+b=b**
se **a=0 e se b=1 0+1=1=a+b=b**
se **a=1 e se b=1 1+1=1=a+b=b**
(a=1 e b=0 non lo posso considerare perche' contrario all'ipotesi)
il che equivale a dire:
$\mathbf{a + b = b}$
come volevamo.

9. Un esempio

Prima di concludere, anche se non servirebbe per informatica, vediamo almeno un esempio classico di Algebra di Boole non binaria ed anche un esempio di insieme parzialmente ordinato.

Dato l'insieme A≡{a,b,c}, consideriamone l'insieme potenza :
℘(A)≡{{∅}, {a}, {b}, {c}, {a,b}, {a,c}, {b,c}, {a,b.c} }
Tale insieme, dotato delle normali operazioni di unione, intersezione e complementare e' un algebra di Boole con **0** che corrisponde all'insieme vuoto **{∅}** ed **1** che corrisponde all' insieme universo **{a,b.c}.**

Vediamolo nei particolari.
Voglio dimostrare che:
(℘(A); ∪, ∩ , ')
e' un algebra di Boole
E' l'operazione di passaggio al complementare e corrisponde alla differenza fra l'insieme universo e l'insieme considerato; cioe' ad esempio:
{a,b}'={c}={a,b,c}\{a,b}
Chiamiamo **x, y, z, ..** dei generici elementi di **℘(A);** e controlliamo che valgono le proprieta'

- ***Legge commutativa***
  x ∪ y = y ∪ x
  a ∩ y = y ∩ x
  Vero per la commutativita' delle operazioni unione ed intersezione fra insiemi.

- ***Legge distributiva***
  x ∪ (y ∩ z) = (x ∪ y) ∩(x ∪ z)
  x ∩(y ∪ z) = (x ∩ y) ∪ (x ∩ z)
  sono le normali proprieta' distributive fra insiemi

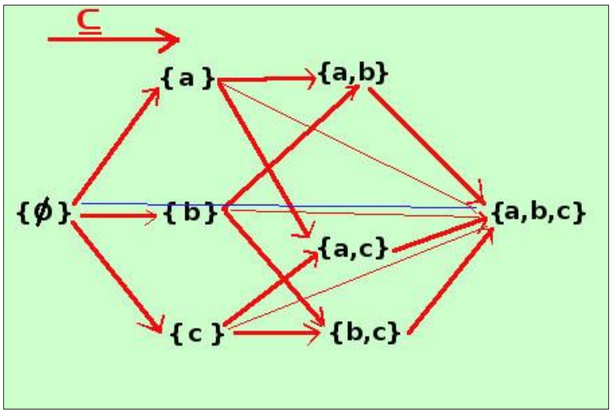
- ***Leggi dell'identita'***
  x ∪ {∅} = x   cioe' **{∅}** corrisponde allo **0**
  x ∩ {a,b.c} = {a,b.c}   cioe' **{a,b.c}** corrisponde ad **1**

- ***Leggi del complemento***
  x ∪ x' = {a,b.c}
  x ∩ x' = {∅}
  Infatti la somma di due complementari da' l'universo e l'intersezione di due complementari da' l'insieme vuoto.

Essendo soddisfatte tutte le proprieta' la struttura **(℘(A); ∪, ∩ , ')** e' un algebra di Boole.

Consideriamo, infine, la normale relazione di inclusione in senso largo **⊆** (contenuto od uguale) fra insiemi.
Diremo che tale relazione e' di ordine parziale su **℘(A)** e che:
{℘(A); ⊆}
e' un insieme parzialmente ordinato.
Infatti, chiamati **x, y, z** elementi qualunque di ℘(A) valgono le 3 proprieta':

- *riflessiva*
  $x \subseteq x \quad \forall x \in \wp(A)$
  ogni insieme e' contenuto od uguale a se' stesso
- *antisimmetrica*
  se $x \subseteq y$ e $y \subseteq x \Rightarrow x=y \quad \forall x,y \in \wp(A)$
- *transitiva*
  $x \subseteq y, y \subseteq z \Rightarrow x \subseteq z \quad \forall x,y,z \in \wp(A)$
  Se il primo iniseme e' contenuto od uguale ad un secondo ed il secondo insieme e' contenuto od uguale ad un terzo ne segue che il primo insieme e' contenuto od uguale al terzo



Tale relazione $\subseteq$, induce sull'insieme $\wp(A)$ un ordine che va dagli insiemi piu' piccoli agli insiemi piu' grandi, e tale relazione e' solo parziale perche' esistono elementi non confrontabili; ad esempio {a} non e' confrontabile con {b,c}.

A destra, una rappresentazione grafica dell'insieme in questione: le linee rosse con verso da sinistra a destra indicano la relazione di inclusione: per ragioni grafiche per l'insieme vuoto (che e' contenuto in tutti gli insiemi) non ho messo le linee di inclusione limitandomi ad una linea azzurra fino all'insieme universo.
Notare la forma a cubo con i vertici corrispondenti agli elementi di $\wp(A)$.

10. Espressioni booleane

a) Introduzione

Come in tutte le algebre che si rispettino (anche se con disperazione degli alunni!), anche qui, nell'algebra binaria di Boole avremo delle possibili espressioni che potranno essere semplificate mediante le regole dell'algebra stessa: questo sara' particolarmente importante perche', essendo la nostra finalita' di costruire dei circuiti elettrici omologhi a tali regole, potremo, prima di procedere alla costruzione, semplificare teoricamente i circuiti fino alla forma piu' semplice con risparmio, poi, di tempo e denaro.

Qui ci limitiamo a dei semplici cenni sulla parte iniziale di tutto il discorso, lasciando agli studiosi di informatica l'approfondimento dell'argomento (semplificazione di circuiti logici) su loro testi specifici.

b) Prodotti fondamentali

Consideriamo un insieme di variabili **x, y, z, t,.....**
Chiameremo **espressione booleana** in tali variabili una qualsiasi espressione costruita da tali variabili utilizzando le operazioni di somma, prodotto e passaggio al complementare.
Esempio: espressione $= x + (y' + x \cdot x' \cdot z)$.
Chiameremo **prodotto fondamentale** un termine oppure un prodotto di piu' termini formato da variabili e/o dai loro complementari ed in cui non compaia piu' di una volta la stessa variabile.
Esempio: sono prodotti fondamentali $x \cdot y, \quad x \cdot y' \cdot z', \quad x' \cdot y' \cdot t$

Non sono prodotti fondamentali $\mathbf{x \cdot x'}$, $\mathbf{x \cdot y' \cdot z' \cdot x}$, $\mathbf{x' \cdot y' \cdot t \cdot t}$
Notiamo che ogni espressione che non sia prodotto fondamentale si puo' ridurre ad un prodotto fondamentale oppure a **0**

Esempio 1

$\mathbf{x \cdot x' = 0}$ (seconda legge del complemento)

Esempio 2

$\mathbf{x \cdot y' \cdot z' \cdot x = y' \cdot x \cdot z' \cdot x =}$ (seconda legge commutativa)
$\mathbf{y' \cdot x \cdot z' \cdot x = y' \cdot z' \cdot x \cdot x}$ (seconda legge commutativa)
$\mathbf{y' \cdot z' \cdot x \cdot x = y' \cdot z' \cdot 0}$ (seconda legge dei confini)
$\mathbf{y' \cdot z' \cdot 0 = y' \cdot 0}$ (seconda legge dei confini)
$\mathbf{y' \cdot 0 = 0}$ (seconda legge dei confini)

Esempio 3

$\mathbf{x' \cdot y' \cdot t \cdot t = x' \cdot y' \cdot t}$ (idempotenza)
In pratica un prodotto fondamentale sara' il pezzo piu' piccolo della nostra espressione booleana non riducibile ulteriormente con le regole del prodotto.

Molto intuitivamente si puo' dire che nelle algebre di Boole stiamo costruendo i monomi delle espressioni come nell'algebra elementare.

Per semplicita', d'ora in avanti, dove non vi siano dubbi tralasciamo il segno del prodotto cioe' invece di scrivere $\mathbf{x \cdot y'}$ scriveremo semplicemente $\mathbf{xy'}$ sottointendendo il $\cdot$

c) Forma normale disgiuntiva

Se nella pagina precedente abbiamo costruito i monomi, qui possiamo dire molto intuitivamente che proviamo a costruire i polinomi; intendiamoci, saranno molto diversi dai polinomi che conosciamo dall'algebra elementare.

Per semplicita', d'ora in avanti, dove non vi siano dubbi, tralasciamo il segno del prodotto cioe' invece di scrivere $\mathbf{x \cdot y'}$ scriveremo semplicemente $\mathbf{xy'}$ sottointendendo il $\cdot$

Consideriamo un'espressione composta dalla somma di uno o piu' prodotti fondamentali, tali che nessuno di essi sia contenuto in un altro; chiameremo tale espressione **forma normale disgiuntiva** od anche **forma di somma-prodotti**.
Esempi di espressione in forma normale disgiuntiva:
Espressione = $\mathbf{xy'z}$
Espressione = $\mathbf{x'y + xyz'}$
Espressione = $\mathbf{xy'z + x'y + xyz'}$
Vediamo anche un esempio di espressione non in forma normale disgiuntiva:
Espressione = $\mathbf{xy'z + x'y + xy'}$
Qui l'ultimo termine $\mathbf{xy'}$ e' contenuto nel primo $\mathbf{xy'z}$
Pero' se faccio:
$\mathbf{xy'z + x'y + xy' = xy'z + xy' + x'y}$ (legge commutativa)
$\mathbf{xy'z + xy' + x'y = (xy'z + xy') + x'y}$ (legge associativa)
$\mathbf{(xy'z + xy') + x'y = (xy' \cdot (1+z)) + x'y}$ (legge distributiva)
$\mathbf{(xy' \cdot (1+z)) + x'y = xy' + x'y}$ (legge dei confini)
il risultato $\mathbf{xy' + x'y}$ e' ora in forma normale disgiuntiva
Cioe' possiamo dire:
*un'espressione booleana diversa da **0** puo' essere sempre messa in forma normale disgiuntiva.*

Per poterlo fare dobbiamo seguire queste regole:

I. *Utilizzando le leggi di de Morgan e del doppio complemento, possiamo spostare l'operazione di complemento verso l'interno delle parentesi fino ad applicarla alle lettere; allora l'espressione sara' formata solamente da somme e prodotti di termini.*
II. *Utilizzando la proprieta'distributiva del prodotto rispetto alla somma, trasformiamo l'espressione in una somma di prodotti, cioe' possiamo eseguire le moltiplicazioni come se fossero "polinomi" ottenendo una somma di "monomi"; e' possibile combinando proprieta' associativa e distributiva del prodotto rispetto alla somma.*
III. *Utilizziamo poi le proprieta' opportune (idempotenza,assorbimento, complemento,....) per trasformare ogni prodotto o in **0** oppure in un prodotto fondamentale.*
IV. *Infine utilizzando la legge dei confini trasformiamo l'espressione in forma normale disgiuntiva.*

Vediamo un esempio.

Esercizio:
trasformare in forma normale disgiuntiva l'espressione:
$((xy')'z)'(x+y')(yz')' =$
Non indico "tutti" i passaggi, tipo la proprieta' commutativa ed associativa per ragioni di spazio, spero che sia chiaro lo stesso il procedimento.

$((xy')'z)'(x+y')(yz')' =$ porto il complementare da fuori a dentro le parentesi piu' esterne (ti ricordo che + diventa · e viceversa)
$= ((xy')'' + z')(x+y')(y' + z'') =$ applico la legge del doppio complemento
$= (xy' + z')(x+y')(y'+z) =$ applico la proprieta' distributiva (moltiplico i primi due)
$= (xy'x + xy'y' + z'x + z'y')(y'+z) =$ leggi dell'idempotenza per il prodotto
$= (xy' + xy'+ z'x + z'y')(y'+z)=$ ancora la legge dell'idempotenza per la somma
$= (xy'+z'x + z'y')(y'+z)=$ applico la proprieta' distributiva (moltiplico)
$= xy'y'+ xy'z +z'xy' + z'xz + z'y'y' +z'y'z =$ applico la legge dell'idempotenza e del complemento
$= xy'+ xy'z +z'xy' + 0x + z'y' + 0y' =$ per la legge dell'identita'
$= xy'+ xy'z +z'xy' + z'y' =$ applico la legge dellassorbimento (xy' e' contenuto in z'xy')
$= xy'z +z'xy' + z'y' =$ applico la legge dellassorbimento (z'y' e' contenuto in z'xy')
$= xy'z +z'xy'$ e questa e' la forma normale disgiuntiva

d) <u>Unicita' della forma normale disgiuntiva completa</u>

Diremo che un'espressione booleana e' in **forma normale disgiuntiva completa** se :

1. e' diversa da **0**
2. e' in forma normale disgiuntiva
3. ogni prodotto e' completo nel senso che contiene tutte le variabili

Se per semplicita' consideriamo solamente tre variabili **x**, **y** e **z,** allora una forma normale disgiuntiva completa puo' essere
espressione $= xyz + x'y'z + xyz'$

Da notare che i termini di un'espressione di questo tipo possono essere al massimo 8; infatti per ogni termine abbiamo due stati (variabile o complementare della variabile) ed i termini sono 3, quindi devo prendere le disposizioni con ripetizione di 2 stati presi 3 a 3 cioe' $2^3 = 8$
I termini possibili sono:
$xyz \quad x'yz \quad xy'z \quad xyz' \quad x'y'z \quad x'yz' \quad xy'z' \quad x'y'z'$

Se sostituiamo variabile **x**, **y**, **z** con **primo posto**, **secondo posto**, **terzo posto** e sostituiamo **0** alla variabile normale ed **1** al complementare otteniamo le possibili terne:

0 0 0   1 0 0   0 1 0   0 0 1   1 1 0   1 0 1   0 1 1   1 1 1

Piu' in generale se abbiamo n variabili allora, avremo 2^n termini possibili.
Abbiamo gia' visto che con 8 variabili (un byte) sono possibili 256 ottuple: $2^8 = 256$

Se un'espressione in forma normale disgiuntiva non e' completa, allora si puo' rendere completa moltiplicando opportunamente il termine cui manca la variabile.
Se, ad esempio, ho il termine **xy'** , per renderlo completo moltiplico per **z+z'**
Infatti per la legge del complemento **z+z'=1** e quindi posso scrivere:
xy' = xy'(z+z') = xy'z + xy'z'

Vale la proprieta'
Ogni espressione booleana diversa da zero puo' essere posta in forma normale disgiuntiva completa e tale rappresentazione e' unica.

Chiudiamo con un esercizio.
Poniamo la seguente espressione booleana in forma disgiuntiva completa:
(x'y)'z = sposto l'operazione di complemento all'interno della parentesi
= (x'' + y')z = doppio complemento
= (x + y')z = moltiplico
= xz+ y'z = questa e' una forma normale disgiuntiva, per renderla completa moltiplico per **1**, cioe' il primo termine per **y+y'** ed il secondo per x+x'
= x(y+y')z+ (x+x')y'z = eseguo le moltiplicazioni
= xyz + xy'z + xy'z + x'y'z = idempotenza (ce ne sono due uguali)
= xyz + xy'z + x'y'z

e) Conclusioni

Anche se l'argomento richiederebbe ben altri approfondimenti, terminiamo qui il discorso teorico sull'algebra binaria di Boole, lasciando agli Informatici uno studio piu' rigoroso ed ampio.
Quindi ora abbiamo un algebra con cui trattare i "pacchetti di dati", abbiamo determinato gli enti su cui lavorare (cioe' i "pacchetti" di 0 ed 1) ora dobbiamo collegare le operazioni viste in algebra (passaggio al complementare, somma, prodotto) a tali enti e trasformare tutto in circuiti elettrici.
Nel prossimo capitolo metteremo in pratica quanto visto in modo teorico, applicando l'algebra a dei fili percorsi o meno da corrente (i due stati, la variabile ed il suo complementare) e collegando tali fatti alle tavole di verita' ed alle operazioni viste in logica per i predicati (and logico, or logico), fino alla costruzione di un semplice circuito che faccia l'operazione di somma fra due bit: la moltiplicazione poi di tale circuito ci permettera' di poter far eseguire le operazioni in modo automatico da una macchina; infine, sostituendo ai numeri dei concetti logici, potremo avere la base di una macchina per elaborare tali concetti (elaboratore elettronico).

D. CIRCUITI LOGICI ED ELETTRICI

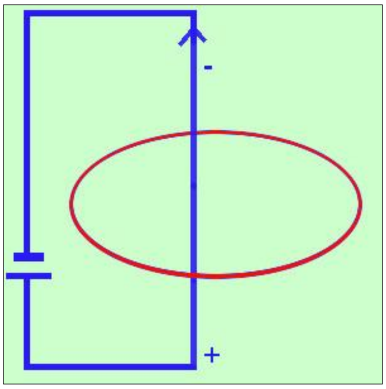
1. Interruttori

a) Il problema

Quello di cui abbiamo bisogno ora e' di un modo pratico per far passare o per interrompere un flusso di corrente su un filo, cioe' un interruttore che la macchina possa utilizzare da sola, intervenendo solamente con flussi di corrente.
Cominceremo dal metodo piu' primitivo, poi vedremo come, col progredire della tecnica, tale metodo si sia evoluto sino ad arrivare a quello odierno.
Questi sarebbero veramente argomenti piu' di fisica che di matematica, ma visto che sono anche stato insegnante di fisica, ne parliamo per sommi capi e semplificando al massimo la narrazione, giusto per avere un'idea dell'andamento del fenomeno.

b) Il circuito interruttore

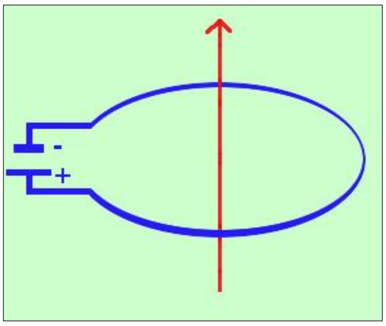
1) Effetto magnetico di una corrente elettrica



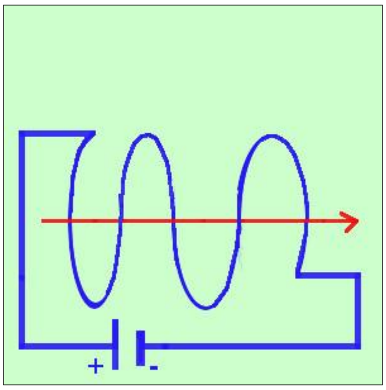
Consideriamo un filo percorso da corrente; se il filo e' collegato ad una pila la corrente per convenzione, parte dal polo positivo (quello piu' lungo e stretto) ed arriva al polo negativo, (quello piccolo e tozzo).
La corrente percorrendo il filo, genera un campo magnetico attorno al filo stesso, cioe' lo spazio attorno al filo, al passaggio della corrente diventa magnetico, come puoi renderti conto avvicinando al filo, ad esempio, una bussola.
Il campo magnetico prende la forma di cilindri attorno al filo stesso.
Esperimento:
prendi un cartoncino forato e fai passare il filo attraverso il foro; tieni il cartoncino orizzontale e versaci della limatura di ferro
Se fai passare corrente sul filo, ad esempio collegandolo ai poli di una batteria elettrica vedrai che la limatura di ferro si dispone a cerchi concentrici con centro il filo.

---

2) Bobina elettrica (solenoide)

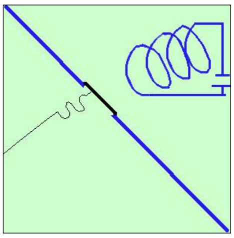


Condideriamo una spira, cioe' un filo circolare (solenoide); se il filo e' collegato ad una pila la corrente genera un campo magnetico che ha maggior concentrazione sull'asse del cerchio, cioe' posso considerare la spira come una piccola calamita.
Se ora prendo piu' spire (bobina elettrica) allora il campo aumenta e l'effetto calamita pure.

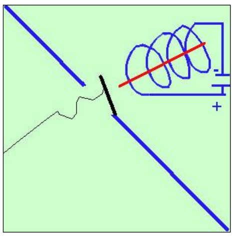


Osservazione: un'applicazione e' l'elettrocalamita che si ottiene mettendo al centro del solenoide una sbarra di ferro dolce; al passaggio della corrente il ferro aumenta l'effetto del solenoide e diventa una calamita attirando oggetti metallici.
Staccando la corrente il ferro si smagnetizza tornando neutro

3) Il circuito interruttore

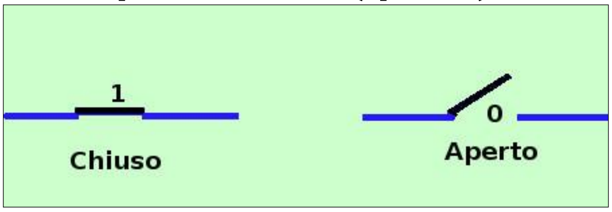


Consideriamo un filo metallico con una lamina mobile tenuta attaccata al filo stesso mediante una molla; allora nel filo passa una corrente.

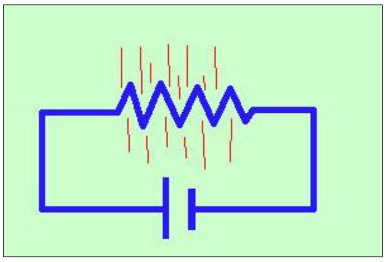


Se ora facciamo passare una corrente elettrica nel solenoide a destra, allora si attiva un campo magnetico che attira la lamina e quindi impedisce il passaggio di corrente nel filo.

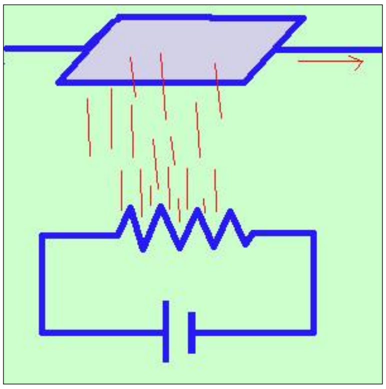
Se poi smetto di erogare corrente al solenoide, allora il campo magnetico sparisce e la molla fa tornare la lamina al suo posto e nel filo puo' ancora passare corrente
Abbiamo quindi un circuito che serve da interruttore per far passare od interrompere il passaggio di corrente in un filo.
Basandoci su questo fatto in futuro indicheremo in questo modo un circuito in cui passi corrente (chiuso **1**) e in cui non passi la corrente (aperto **0**)



c) Il diodo



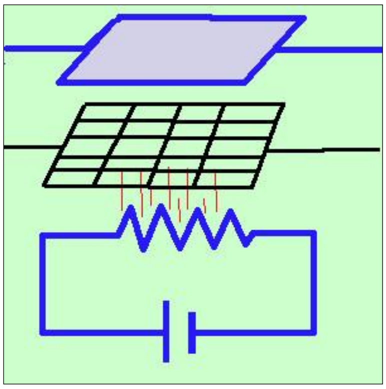
Il diodo e' un circuito piu' elaborato che puo' servire come interruttore per far passare o meno la corrente elettrica in un circuito.



Si basa sull'effetto termoionico:
se in un circuito facciamo passare una corrente elettrica e vi mettiamo una resistenza allora il circuito si scalda ed emette elettroni .

se ora poniamo di fronte alla resistenza una placca metallica e la carichiamo positivamente gli elettroni emessi dalla resistenza si dirigeranno verso la placca e si avra' una corrente (la corrente elettrica e' movimento di elettroni) e nel circuito superiore fluisce una corrente.

Se interponiamo fra il flusso di elettroni e la lamina metallica una griglia, se la griglia e' caricata negativamente, allora il suo campo elettrico bloccherа' il flusso di elettroni, mente, se e' neutra ne permettera' il passaggio: quindi fornendo o meno corrente alla griglia apro e chiudo il circuito ed ho un interruttore.
Il diodo, costruito in un tubo a vuoto, e' una valvola e il suo utilizzo ha permesso di costruire i primi computer: l'ingombro di una valvola, pur se limitato, obbligava pero' a fare computer delle dimensioni di un appartamento.

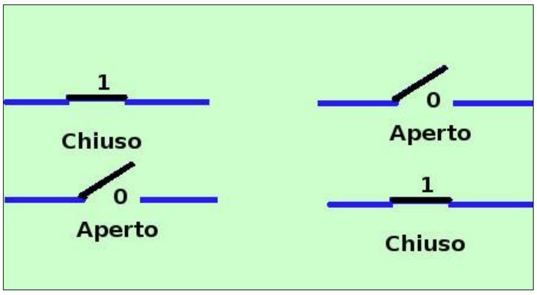


d) Transistor e chip di silicio

Attorno agli anni 50 fu inventato il transistor, che si puo' comportare come un diodo, e quindi come un interruttore, le cui dimensioni, pero', rispetto alle valvole sono notevolmente ridotte: questo ha permesso di passare da computer delle dimensioni di un appartamento a computer delle dimensioni di una stanza
Ricordo che nel 1959 comperai la mia prima radiolina a transistor (7 transistor!): il transistor aveva permesso la costruzione di una radiolina portatile al posto delle radio a valvole che erano parecchio ingombranti e questo permetteva di portarsi dietro ovunque la radiolina un po' come fate voi oggi con i telefonini
Attorno agli anni 60 poi, si e' passati ai chip di silicio: lenti di silicio purissimo venivano trattate con strati di sostanze speciali e poi tali strati venivano asportati su linee a varie profondita' costituendo circuiti, diodi, ed altri apparati elettrici in miniatura: i computer divennero delle dimensioni di un tavolo
infine negli anni 70, con i microchip, chip di silicio ove i componenti venivano lavorati a scale microscopiche, il computer arrivo' alle dimensioni attuali e nei primi anni 80 nasceva il primo personal computer e nel 1984 anche un computer dotato di interfaccia grafica e mouse
Comunque, chiudendo l'argomento, abbiamo un interruttore che possiamo far aprire o chiudere da una corrente elettrica e questo, per i nostri circuiti, ci e' sufficiente

2. Operazione unaria: il passaggio al complementare

Possiamo pensare l'attivazione di un interruttore come il passaggio al complementare nell'algebra di Boole: infatti se lo stato e' **1** aprendo l'interruttore passeremo allo stato **0** e se lo stato e' **0** chiudendo l'interruttore passeremo allo stato **1**

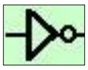


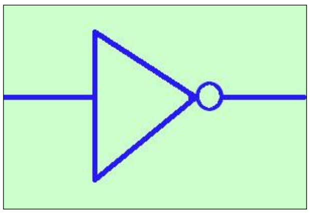
a	a'
1 0	0 1

Questo, sostituendo **0** con **FALSO** ed **1** con **VERO** corrisponde alla tavola di verita' per la negazione logica.
Nell'algebra di Boole posso chiamare gli elementi indifferentemente **0** ed **1** oppure **F** e **V** e questo ci permettera' di usare il computer oltre che per fare calcoli numerici anche per fare calcoli logici, con tutte le possibilita' che cio' offre :

a	a'
p	p _
V F	F V

Tale circuito in informatica viene detto **porta not** o semplicemente **not** ed e' tale che il valore in ingresso viene cambiato in uscita nel suo valore complementare;cioe' se entra **1** esce **0** e se entra **0** esce **1**

per indicarla si usa il simbolo 

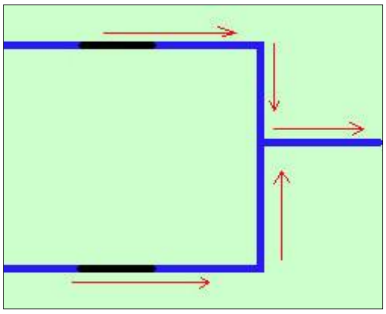


In qualche testo c'e'un cerchietto all'uscita, in qualche altro no, noi lo useremo preferibilmente con il cerchietto, perche' il numero di testi con cerchietto e' superiore al numero di quelli senza:
Non so di preciso perche' c'e' il cerchietto e a cosa serva: su internet non ho trovato nulla.
Penso sia usanza mettere il cerchietto in quelle porte che "invertono" il risultato, ma e' solo una deduzione. Se qualcuno di voi lo sa e me lo fa sapere mi fara' un grosso favore: l'indirizo e-mail e' nella home page; grazie

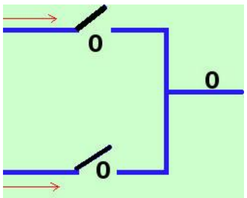
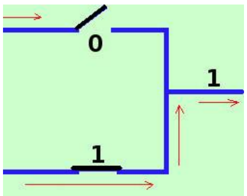
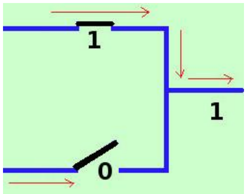
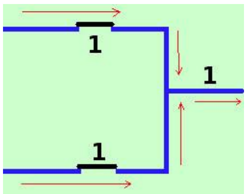
3. Prima operazione binaria: la somma logica

La somma in un'algebra di Boole puo' essere pensata come una coppia di interruttori in parallelo, cioe' tali che la corrente possa percorrere il primo ed il secondo cammino prima di ricongiungersi

+	0	1
0	0	1
1	1	1



infatti indicando con **1** il passaggio di corrente e con **0** il non passaggio avremo

$0 + 0 = 0$	
$0 + 1 = 1$	
$1 + 0 = 1$	
$1 + 1 = 1$	

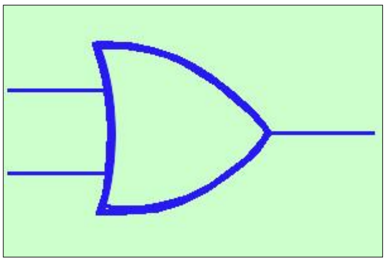
Questo, sostituendo **0** con **FALSO** ed **1** con **VERO** corrisponde alla tavola di verita' per la disgiunzione inclusiva .

Ti ricordo che, nell'algebra di Boole, posso chiamare gli elementi indifferentemente **0** ed **1** oppure **F** e **V** e questo ci permettera' di usare il computer oltre che per fare calcoli numerici anche per fare calcoli logici, con tutte le possibilita' che cio' offre

---

a	b	a + b
p	q	p V q
f	f	f
f	v	v
v	f	v
v	v	v

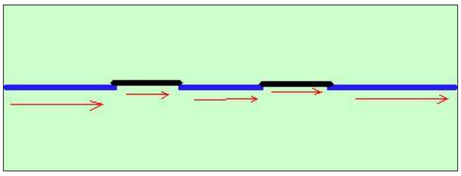
Tale circuito in informatica viene detto **porta logica or** o semplicemente **or** ed e' tale che il valore in uscita e' **0** solamente se entrambe gli ingressi sono **0**
cioe' per avere l'uscita **1** deve essere **1** o il primo ingresso o il secondo oppure tutti e due
per indicarla si usa il simbolo 



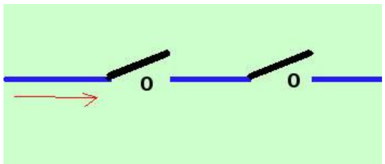
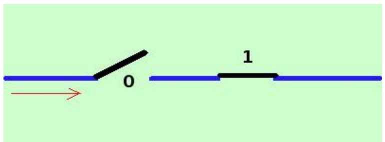
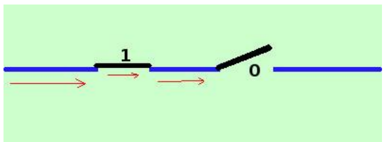
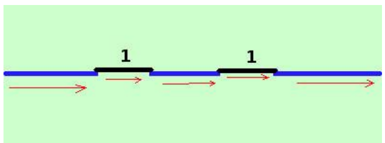
4. Seconda operazione binaria: il prodotto logico

Il prodotto in un'algebra di Boole puo' essere pensato come una coppia di interruttori in serie, cioe' tali che la corrente percorrera il primo e, successivamente il secondo cammino prima di proseguire nel circuito

·	0	1
0	0	0
1	0	1



infatti indicando con **1** il passaggio di corrente e con **0** il non passaggio avremo

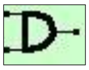
$0 \cdot 0 = 0$	
$0 \cdot 1 = 0$	
$1 \cdot 0 = 0$	
$1 \cdot 1 = 1$	

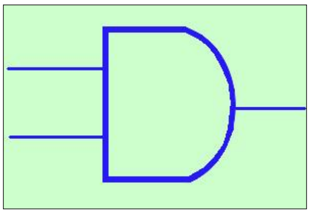
Questo, sostituendo **0** con **FALSO** ed **1** con **VERO** corrisponde alla tavola di verita' per la congiunzione logica.

Ti ricordo ancora una volta che, nell'algebra di Boole, posso chiamare gli elementi indifferentemente **0** ed **1** oppure **F** e **V** e questo ci permettera' di usare il computer oltre che per fare calcoli numerici anche per fare calcoli logici, con tutte le possibilita' che cio' offre.

---

a	b	$a \cdot b$
p	q	$p \wedge q$
f	f	f
f	v	f
v	f	f
v	v	v

Tale circuito in informatica viene detto **porta logica and** o semplicemente **and** ed e' tale che il valore in uscita e' **1** solamente se entrambe gli ingressi sono **1.** Per indicarla si usa il simbolo 



5. Altre porte logiche

a) Introduzione

Possiamo dire che, con le tre porte logiche che abbiamo visto, possiamo considerare completo l'elenco delle porte logiche fondamentali, perche' ogni altra porta e' costruita partendo da queste, in linea con quello che avevamo gia' visto in logica dove le tre operazioni **not, vel, et** ci permettevano di costruire tutte le altre operazioni logiche che potevamo considerare.

Pero' per semplificare, nella costruzione di circuiti, e' spesso utile accorpare piu' porte in una sola, guadagnandone in chiarezza se non in semplicita', e siccome sono parecchio usate, vediamo le principali altre porte che vengono utilizzate in informatica.

Prendo l'occasione per far notare ancora una volta la stretta corrispondenza esistente fra logica ed informatica che mostra come le due discipline siano aspetti diversi della stessa realta'.
Studiamo quindi le altre porte, anche per mostrarne il parallelo con le operazioni gia' fatte in logica.

b) Porta nand

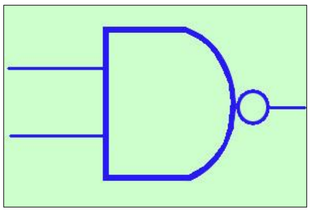
Consideriamo la composizione di una porta and con una porta not; la sua uscita e' il contrario della porta and cioe' indicando con **1** il passaggio di corrente e con **0** il non passaggio avremo:

a	b	a · b	(a · b)'
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

Ricordo ancora che il prodotto (la somma indicata nella pagina successiva) non ha nulla a che fare con le normali operazioni di prodotto (di somma) ma e' solo un'operazione definita nell'algebra binaria di Boole che indichiamo con · (+) solo per comodita' .

Questo, sostituendo 0 con FALSO ed 1 con VERO, corrisponde alla tavola di verita' di negazione della congiunzione logica

a	b	a · b	(a · b)'
p	q	p ∧ q	p ∧ q___
f	f	f	v
f	v	f	v
v	f	f	v
v	v	v	f



Tale circuito in informatica viene detto porta logica nand o semplicemente nand ed e' tale che il valore in uscita e' **0** solamente se entrambe gli ingressi sono **1.**
Per indicarla si usa il simbolo qui a destra (notare il tondino all'uscita che significa la negazione)

---

c) Porta nor

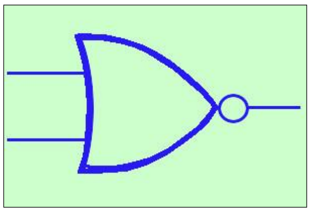
Consideriamo la composizione di una porta or con una porta not; la sua uscita e' il contrario della porta or, cioe' indicando con **1** il passaggio di corrente e con **0** il non passaggio avremo:

a	b	a + b	(a + b)'
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

Ricordo ancora che la somma (il prodotto indicato nella pagina precedente) non ha nulla a che fare con le normali operazioni di somma (di prodotto) ma e'solo un'operazione definita nell'algebra binaria di Boole che indichiamo con **+** (·) solo per comodita' .

Questo, sostituendo **0** con **FALSO** ed **1** con **VERO** corrisponde alla tavola di verita' di negazione della disgiunzione inclusiva

a	b	a + b	(a + b)'
p	**q**	**p ∨ q**	$\overline{\mathbf{p} \vee \mathbf{q}}$
f	f	f	v
f	v	v	f
v	f	v	f
v	v	v	f



Tale circuito in informatica viene detto **porta logica nor** o semplicemente **nor** ed e' tale che il valore in uscita e' **1** solamente se entrambe gli ingressi sono **0.**
Per indicarla si usa il simbolo qui a destra (notare il tondino

all'uscita che significa la negazione)

d) Porte logiche come disposizioni con ripetizione

Prima di procedere diamo uno sguardo d'insieme alle possibili porte logiche che derivano da due fili percorsi o meno da corrente. Indicando con **0** il non passaggio di corrente e con **1** il passaggio di corrente notiamo che si tratta di disposizioni con ripetizione di 2 oggetti (**0** e **1**) presi 4 a 4, cioe' con 2 proposizioni avremo per le porte logiche 16 possibilita' ($\mathbf{D'}_{2,4} = \mathbf{2^4=16}$)

La stessa cosa abbiamo fatto in logica: se vuoi Vedere i 16 possibili casi in logica

---

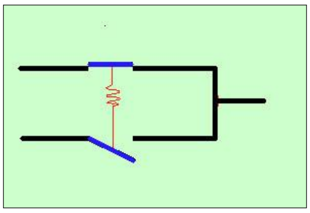
Nella seguente tabella elenco le 16 possibilita'; chiamo
a primo filo
b secondo filo

a	b	T	a+b	a+b'	a'+b	(a·b)'	a	b	(a·b')+(a'·b)	(a·b)+(a'·b')	b'	a'	a·b	a·b'	a'·b	(a+b)'	C
0	0	1	0	1	1	1	0	0	0	1	1	1	0	0	0	1	0
0	1	1	1	0	1	1	0	1	1	0	0	1	0	0	1	0	0
1	0	1	1	1	0	1	1	0	1	0	1	0	0	1	0	0	0
1	1	1	1	1	1	0	1	1	0	1	0	0	1	0	0	0	0

NOME DEI CIRCUITI PRESENTI IN CIASCUNA COLONNA

T (1,1,1,1) - Tautologia

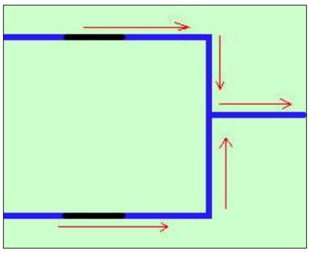
E' la proprieta' per cui il filo e' sempre percorso da corrente; comunque siano le funzioni componenti possiamo rappresentarlo con un **1.** Un circuito potrebbe essere:



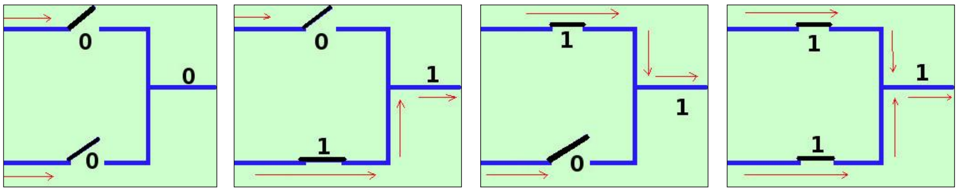
La "molla" in mezzo fa in modo che se un circuito e' chiuso l'altro e' aperto e non permette che i due circuiti possano essere chiusi contemporaneamente, quindi avremo sempre il passaggio di corrente.
In forma normale disgiuntiva completa possiamo pensarla come:
1= a'b'+a'b+ + ab'+ab
Per esercizio, dimostriamolo algebricamente:
1 = 1·(a+a') = moltiplico per **(a+a')=1** per la prima legge del complemento
= a + a' = (a+a')(b+b')= moltiplico per **(b+b')=1** per la prima legge del complemento
=ab + ab' + a'b + a'b'
ed ordinando otteniamo, come avevamo gia' visto nella pagina della tabella:
1 = a'b' + a'b + ab' + ab

a+b (0,1,1,1) - Somma

E' la somma definita nell'algebra di Boole **a+b**
corrisponde alla disgiunzione inclusiva in logica
Possiamo rappresentarlo con il circuito



Infatti abbiamo le 4 possibilita':



In forma normale disgiuntiva completa possiamo pensarla come:
$a+b = a'b + ab' + ab$
Per esercizio dimostriamolo algebricamente:
$a+b = (a+b)(a+a') =$ moltiplico per $(a+a')=1$ per la prima legge del complemento
$= aa + aa' + ab + ab' =$ sviluppo; $aa=a$ per la seconda legge dell'idempotenza e $aa'=0$ per la seconda legge del complemento
$= a + ab + a'b =$ so che $a+ab=a$ per la prima legge di assorbimento
$= a + a'b =$
$= a(b+b')+a'b =$ moltiplico per $(b+b')=1$ il primo termine ed ottengo
$= ab + ab' + a'b$
cioe' ordinando come avevamo gia' visto nella pagina della tabella
$a+b = a'b + ab' + ab$

a+b' (1,0,1,1) - Implicazione diretta

E' L'implicazione diretta $a + b'$; corrisponde alla implicazione diretta in logica
In forma normale disgiuntiva completa possiamo pensarla come:
$a+b' = ab + ab' + a'b'$
Per esercizio dimostriamolo algebricamente:
$a+b' = (a+b')(a+a') =$ moltiplico per $(a+a')=1$ per la prima legge del complemento
$= aa + aa' + ab' + a'b' =$ sviluppo; $aa=a$ per la seconda legge dell'idempotenza e $aa'=0$ per la seconda legge del complemento
$= a + ab' + a'b' =$ so che $a+ab'=a$ per la prima legge di assorbimento
$= a + a'b' = a(b+b')+a'b' =$ moltiplico per $(b+b')=1$ il primo termine ed ottengo
$= ab + ab' + a'b'$
come volevamo.

a'+b (1,1,0,1) - Implicazione inversa

E' L'implicazione inversa $a' + b$. Corrisponde alla implicazione materiale in logica
In forma normale disgiuntiva completa possiamo pensarla come:
$a'+b= a'b'+a'b+ab$
Per esercizio dimostriamolo algebricamente:
$a'+b = (a'+b)(a+a') =$ moltiplico per $(a+a')=1$ per la prima legge del complemento
$= a'a + a'a' + ab + a'b =$ sviluppo; $a'a'=a'$ per la seconda legge dell'idempotenza e $aa'=a'a=0$ per la seconda legge del complemento
$= a' + ab + a'b =$ so che $a'+a'b=a'$ per la prima legge di assorbimento
$= a' + ab = a'(b+b')+ab =$ moltiplico per $(b+b')=1$ il primo termine ed ottengo
$= a'b + a'b' + ab =$
Ordino ed ottengo:
$= a'b' + a'b + ab$
come volevamo.

(a · b)' (1,1,1,0) - Disgiunzione inversa

E' la disgiunzione inversa o incompatibilita' $(ab)'$. Corrisponde alla disgiunzione inversa in logica
In forma normale disgiuntiva completa possiamo pensarla come:
$(ab)'= a'b'+a'b+ab'$
Per esercizio dimostriamolo algebricamente:

(ab)' = a'+b' = per il principio di dualita'
= a'+b' = (a'+b')(a+a') = moltiplico per **(a+a')=1** per la prima legge del complemento
= a'a + a'a' + ab' +a'b' = sviluppo; **a'a'=a'** per la seconda legge dell'idempotenza e **a'a=0** per la seconda legge del complemento
= a' + ab' + a'b' = so che **a'+a'b'=a'** per la prima legge di assorbimento
= a' + ab' = a'(b+b')+a'b' = moltiplico per **(b+b')=1** il primo termine ed ottengo
= a'b + a'b' + a'b'
come volevamo.

a (0,0,1,1) - Indipendenza inversa

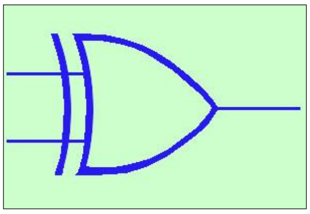
E' l'indipendena inversa o incompatibilita' **a**. Corrisponde alla indipendenza inversa in logica
In forma normale disgiuntiva completa possiamo pensarla come:
a= ab'+ ab
Per dimostrarlo algebricamente basta moltiplicare per **(b'+b)=1** e, per la prima legge del complemento, otteniamo:
a = a (b'+b) = ab' +ab =
come volevamo.

b (0,1,0,1) - Indipendenza diretta

E' l'indipendena diretta **b**. Corrisponde alla indipendenza diretta in logica .
In forma normale disgiuntiva completa possiamo pensarla come:
b= a'b+ ab
Per dimostrarlo algebricamente basta moltiplicare per **(a'+a)=1** e, per la prima legge del complemento, otteniamo:
b = b (a'+a) = a'b + ab =
come volevamo.

(a · b')+(a' · b) (0,1,1,0) – Contro-omologia

E' la contro-omologia **ab'+a'b**. Corrisponde alla disgiunzione esclusiva in logica : e' **1** se uno dei due componenti e' uno , ma non entrambe.
E' gia' in forma normale disgiuntiva completa ,
E' una porta molto importante perche' facendo corrispondere alla somma fra due unita' lo zero sara' la porta che useremo nella costruzione del circuito per fare la somma normale (quella solita, non quella dell'algebra di Boole) fra due numeri binari.
Viene di solito chiamata porta **xor** (or esclusivo perche' vale 1 solamente se e' 1 il primo componente, oppure il secondo ma non entrambi) ed e' rappresentata con il simbolo qui a fianco disegnato.



(a · b)+(a' · b') (1,0,0,1) - Omologia

E' l'omologia **ab+a'b'**. Corrisponde alla coimplicazione in logica : e' **1** solamente se entrambe i componenti sono uguali.
E' gia' in forma normale disgiuntiva completa

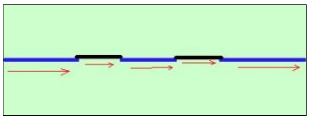
b' (1,0,1,0) - Contro-indipendenza diretta

E' la contro-indipendenza diretta **b'**. Corrisponde alla contro-indipendenza diretta in logica.
In forma normale disgiuntiva completa possiamo pensarla come:
b'= a'b'+ ab'
Per dimostrarlo algebricamente basta moltiplicare per **(a'+a)=1** e, per la prima legge del complemento, otteniamo:
b' = b' (a'+a) = a'b' + ab' =
come volevamo.

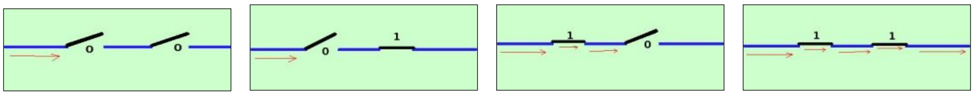
a' (1,1,0,0) - Contro-indipendenza inversa

E' la contro-indipendenza inversa **a'**. Corrisponde alla contro-indipendenza inversa in logica.
In forma normale disgiuntiva completa possiamo pensarla come:
a'= a'b'+ a'b
Per dimostrarlo algebricamente basta moltiplicare per **(b'+b)=1** e, per la prima legge del complemento, otteniamo:
a' = a' (b'+b) = a'b' + a'b =
come volevamo.

ab (0,0,0,1) - Prodotto
E' il prodotto **ab**. Corrisponde alla congiunzione diretta in logica .
Possiamo rappresentarlo con il circuito:



Infatti abbiamo le 4 possibilita':



E' gia' in forma normale disgiuntiva completa

ab' (0,0,1,0) - Contro-implicazione diretta
E' la contro-implicazione diretta **ab'**. Corrisponde alla contro-implicazione diretta in logica.
E' gia' in forma normale disgiuntiva completa.

a'b (0,1,0,0) - Contro-implicazione inversa
E' la contro-implicazione inversa **a'b**. Corrisponde alla contro-implicazione inversa in logica .
E' gia' in forma normale disgiuntiva completa

(a+b)' (1,0,0,0) - Congiunzione inversa
E' la congiunzione inversa **(a+b)'**. Corrisponde alla congiunzione inversa in logica .
(a+b)' = a'·b' per il principio di dualita'
a'b' e' in forma normale disgiuntiva completa.

C (0,0,0,0) - Identita' zero
E' l'identita' zero. Corrisponde alla contraddizione in logica
Non e' rappresentabile

Da notare che applicando le leggi di dualita' e' possibile dare piu' "etichette" ad una stessa colonna; alcune "etichette" sono in forma normale disgiuntiva completa, tipo ad esempio:
(a·b')+(a'·b)
altre no, ad esempio:
(a+b)' non e' in forma disgiuntiva completa.

Importante!

Osserva nelle prime due colonne (quelle colorate) **a** e **b** e considera gli **1** come variabili e gli **0** come i loro complementari.
Allora hai che in ogni colonna ottieni la porta scritta nella forma normale disgiuntiva.

Infatti in ogni colonna:
il primo termine corrispondera' ad **a'b' (a=0, b=0)**
il secondo termine corrispondera' ad **a'b (a=0, b=1)**
il terzo termine corrispondera' ad **ab' (a=1, b=0)**
il quarto termine corrispondera' ad **ab (a=1, b=1)**

Infatti:
nella tautologia hai **1, 1, 1, 1** che puoi tradurre come **a'b'+a'b+ ab'+ab**
nella somma **a+b** hai **0,1,1,1** che puoi tradurre come **a'b+ab'+ab**

nella implicazione diretta **a+b'** hai **1,0,1,1** che puoi tradurre come **a'b'+ab'+ab**

....

La contraddizione (ultima colonna) non ha rappresentazione; cioe':

Nelle porte logiche a due ingressi la forma normale disgiuntiva rappresenta tutti gli stati possibili degli ingressi di un circuito che danno come uscita il valore 1.

Non so se lo stesso risultato sia valido anche per circuiti a piu' ingressi.

Comunque per alcune porte (per esercizio) troveremo la forma normale disgiuntiva completa anche in modo algebrico.

Da notare anche che per ogni porta esiste la porta complementare (basta che nella colonna vengano scambiati **0** ed **1**).

Tutto questo con **2** ingressi **a** e **b**, cioe'

$2^{2^2} = 2^4 = 16$ circuiti possibili

Se avessi **3** ingressi **a**, **b** e **c** allora avrei

$2^{2^3} = 2^8 = 256$ circuiti possibili

Con **4** ingressi **a**, **b**, **c** e **d** avrei

$2^{2^4} = 2^{16} = 65.536$ circuiti possibili

....

6. Equivalenza delle operazioni in informatica, logica e teoria degli insiemi

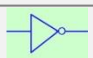
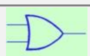
Vediamo ora di puntualizzare che le operazioni fondamentali sugli insiemi, sulle proposizioni logiche, nell'algebra di Boole ed in informatica sono esattamente lo stesso tipo di operazioni, il che fa pensare che il ragionamento matematico e' unico, anche se variano gli oggetti cui si applica

A tal proposito Hilbert diceva che la matematica sarebbe comunque la stessa se invece di operare sui numeri operasse su boccali di birra.

Le operazioni fondamentali sono le operazioni di base da cui derivano tutte le altre.

Esse sono:

- nella teoria degli insiemi: il passaggio al complementare (**A_**), l'unione (**∪**) e l'intersezione (**∩**)
- in logica:la negazione logica (**not**), la disgiunzione inclusiva (**vel** ∨) e la congiunzione logica (**and** ∧)
- nell'algebra di Boole: la negazione (**a'**), la somma (**a+b**) ed il prodotto (**a·b**)
- in informatica: la negazione (porta **not**), la somma (porta **or**) ed il prodotto (porta **and**)

Ambito			
Insiemistica	**Complementare $\overline{A}$**	**Unione ∪**	**Intersezione ∩**
Logica	**Negazione logica $\overline{p}$**	**Disgiunzione inclusiva ∨**	**Congiunzione logica ∧**
Algebra di Boole	**Complementare a'**	**Somma a+b**	**Prodotto a·b**
Informatica	**Porta not** 	**Porta or** 	**Porta and** 

7. Il circuito somma binaria

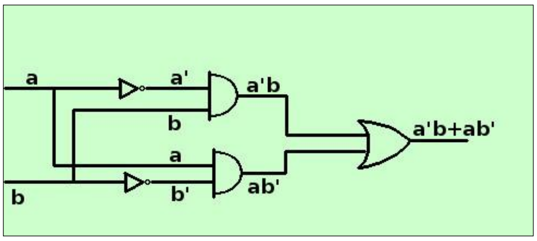
Terminiamo questo capitolo mostrando come costruire il circuito somma binaria, che e' alla base di tutte le operazioni matematiche e logiche che e' possibile fare con il computer. Il nostro problema e' trasformare la somma nell'algebra binaria di Boole in modo che, tramite un particolare circuito, diventi la somma fra due numeri binari:

somma in algebra di Boole	somma binaria
$0+0=0$	$0+0=0$
$0+1=1$	$0+1=0$
$1+0=1$	$1+0=0$
$1+1=1$	$1+1=10$

In pratica si tratta di modificare l'ultima riga in modo da avere il "riporto" nell'operazione binaria; togliendo il riporto l'operazione e' identica alla porta logica **xor** (or esclusivo) ,

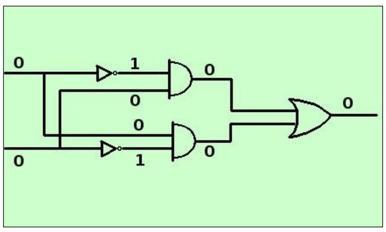
porta xor	
0	$0 \rightarrow 0$
0	$1 \rightarrow 1$
1	$0 \rightarrow 1$
1	$1 \rightarrow 0$

Bastera' quindi modificare leggermente il circuito di tale porta per avere il risultato cercato.
La porta **xor** e' caratterizzata dalla forma normale disgiuntiva completa **ab'+a'b** cioe' dal circuito logico (dove i fili si incrociano non c'e' contatto ma vi sono dei ponti) :

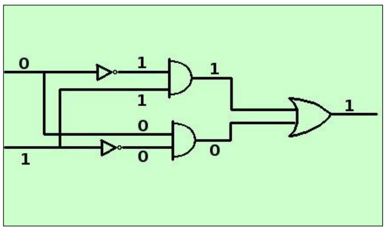
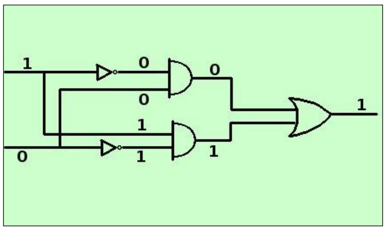
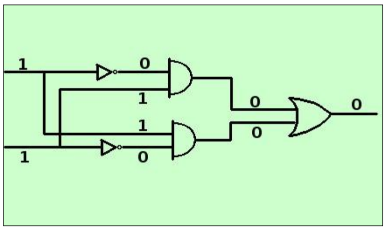


Se vuoi vedere i vari casi possibili

I casi possibili sono:

- 

$0 \quad 0 \rightarrow 0$

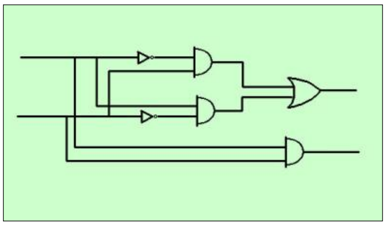
-  $0\ \ 1 \to 1$
-  $1\ \ 0 \to 1$
-  $1\ \ 1 \to 0$

Cioe'

input a	input b	porta xor
0	0	0
0	1	1
1	0	1
1	1	0

Come volevamo.

Aggiungiamo il "riporto" nel seguente modo ed otteniamo il circuito desiderato:

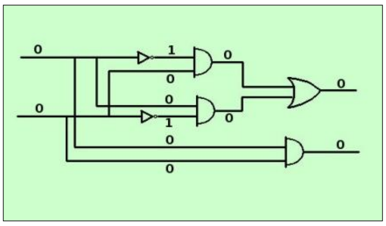
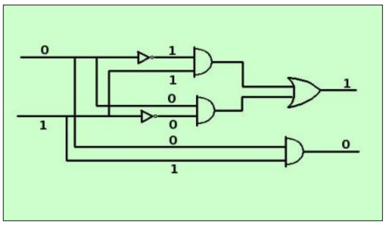
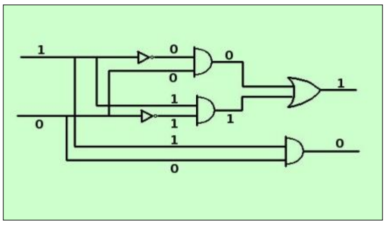
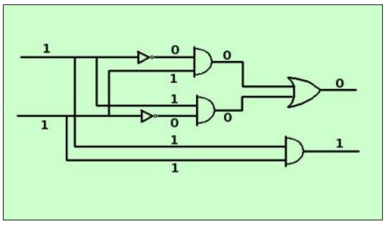


Infatti ora abbiamo:

somma binaria
$0+0=00$
$0+1=01$
$1+0=01$
$1+1=10$

Se vuoi vedere i vari casi possibili

I casi possibili sono :

-  **0 + 0 = 00**
-  **0 + 1 = 01**
-  **1 + 0 = 01**
-  **1 + 1 = 10**

Cioe':

input a	input b	somma
0	0	00
0	1	01
1	0	01
1	1	10

Come volevamo.

Ora il prodotto e' una somma ripetuta; la differenza si puo' pensare come somma complementare ed il quoziente e' una differenza ripetuta; quindi questo sara' il circuito modulare che, ripetuto, ci permettera' di impostare sulla macchina le operazioni matematiche (ma anche le operazioni logiche).

8. Conclusioni

Abbiamo visto che, utilizzando il sistema binario e l'algebra binaria di Boole, e' possibile far eseguire operazioni di somma a due correnti elettriche che percorrano un particolare circuito e che tali operazioni si possono estendere alla logica con tutto quello che cio' comporta; ora, avendo acquisito gli strumenti matematici, si tratta di elaborare l'architettura del calcolatore elettronico con la sua memoria e la sua unita' operativa, ma questo e' specifico dell'informatica
Terminiamo qui i cenni sulla parte relativa ai circuiti, ricordando che ci siamo prefissi di limitarci alla parte matematica e quindi, per approfondimenti ed argomenti specifici propri e' meglio rivolgersi a siti di informatica.